(Part 1 of 2 Parts)

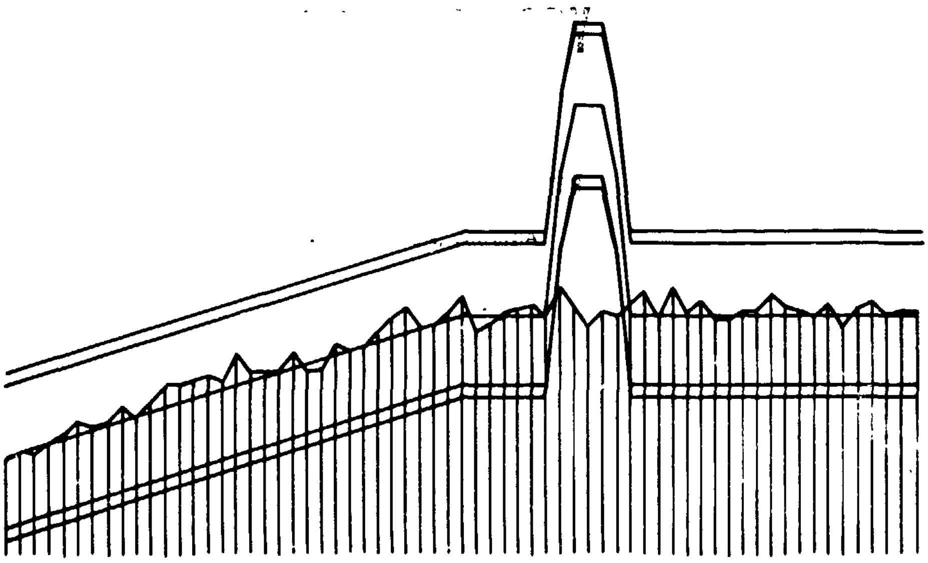
SEMINAR ON UNDERSTANDING DIGITAL CONTROL AND ANALYSIS IN VIBRATION TEST SYSTEMS

Jointly Sponsored By

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

JET PROPULSION LABORATORY
Pasadena, California

THE SHOCK AND VIBRATION INFORMATION CENTER
Naval Research Laboratory, Washington, D.C.



A Publication of
THE SHOCK AND VIBRATION
INFORMATION CENTER
Naval Research Laboratory, Washington, D.C.

Approved for public release; distribution unlimited.

SEMINAR MANAGEMENT

Edward Kirchman, Goddard Space Flight Center
Fredric van Biene, Jet Propulsion Laboratory
Henry C. Pusey, Shock and Vibration Information Center

Program Committee

Robert A. Dorian, Goddard Space Flight Center
Phillip Chapman, Jet Propulsion Laboratory
David Smallwood, Sandia Corporation
Rudolph H. Volin, Shock and Vibration Information Center

Arrangements

Edmund Fitch, Goddard Space Flight Center
Raymond L. Cadwallader, Jet Propulstion Laboratory

Publicity

David Orbock, Northrop Services Inc.

Proceedings

Graphic Arts Branch, Technical Information Division
Naval Research Laboratory

SEMINAR ON UNDERSTANDING DIGITAL CONTROL AND ANALYSIS IN VIBRATION TEST SYSTEMS

MAY 1975

A Publication of
THE SHOCK AND VIBRATION
INFORMATION CENTER
Naval Research Laboratory, Washington, D.C.

The Seminar was held at the Goddard Space Flight Center, Greenbelt, Maryland, on 17-18 June 1975 and at the Jet Propulsion Laboratory, Pasadena, California, on 22-23 July 1975.

CONTENTS

Tutorial Papers

DIGITAL VIBRATION CONTROL TECHNIQUES 1
P. Chapman

MINICOMPUTERS 73
Stephen A. Kallis, Jr.

DIGITAL ANALYSIS AND CONTROL IN THE VIBRATION LABORATORY 79
Peter Moseley

DIGITAL VIBRATION CONTROL TECHNIQUES[1]

P. Chapman

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

ABSTRACT

Analog vibration control techniques are reviewed and are compare with digital techniques. The advantages of the digital methods over the analog methods are demonstrated. The following topics are covered:

(1) Methods of computer-controlled random vibration and reverberation acoustic testing.

(2) Methods of computer-controlled sinewave vibration testing.

(3) Methods of computer-controlled shock testing.

Basic concepts are stressed rather than specific techniques or equipment. General algorithms are described in the form of block diagrams and flow diagrams. Specific problems and potential problems are discussed. The material is computer sciences oriented but is kept at a level that facilitates an understanding of the basic concepts of computer-controlled induced environmental test systems. An introduction to computers is included as an appendix for a review of computer technology as it exists in early 1975.

I. INTRODUCTION AND GENERAL BACKGROUND

Before the introduction of computer-controlled test systems for induced vibration environments, industry used only closed-loop analog techniques for environmental vibration test control. Figure 1 is a block diagram of a conventional closed-loop vibration analog test system. A power amplifier is used to drive an electrodynamic vibration exciter (shaker). Accelerometers or other transducers mounted on a test fixture or test specimen send ac signals to the conditioning or charge amplifiers. The output signals from the charge amplifiers are proportional to the acceleration at the accelerometer mounting points. The signals pass through analog servo control electronics, and this analog electronic device continuously corrects for the response spectrum by correcting the input level and frequency to the desired test specification.

With the advent of the first hardwired special-purpose computers for signal processing, digital control of induced environmental testing was seriously considered. It was the development of these analyzers that has led to computer-controlled testing. The first computer-controlled test system was an attempt to control random-type vibration tests.

The vibration test laboratory has been a natural place to apply automated digital signal processing and control techniques. The highly specialized analog instruments that have traditionally been used are generally limited in performance, difficult to set and operate, expensive to maintain, and are prone to early obsolescence. The first attempts to overcome some of these deficiencies with digital techniques used large digital computers. Technical analysis performance was much improved, and the cost of processing large amounts of data was decreased significantly, but other problems were introduced. The centralized data processing facility is inherently remote from the vibration test laboratory and must be manned by an army of specialists. To keep costs reasonable, data must be processed off-line, often resulting in turnaround times of 2 or 3 days. The remoteness and the off-line processing requirement prevent the operator from monitoring and interacting with the test while it is being performed. Attempts have been made to overcome these deficiencies with use of remote time-shared terminals and high-rate data transmission links. The high cost of these techniques limits their use to the very large test facilities.

The next generation of signal processing equipment, first available commercially about six years ago, was essentially a special-purpose hardwired computer dedicated to performing specific functions rapidly and economically. The cost was low enough

[1] This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract NAS 7-100, sponsored by the National Aeronautics and Space Administration.

and the equipment small enough so that it could be put back in the laboratory under direct control of the test operator. Specialized computer operators were not required because the instruments had a familiar type of control console, minimizing the need for the operator to understand the intricacies of digital processing. The hard-wired approach was necessary to achieve the required speed of processing economically, but had the disadvantage of being highly specialized and not flexible. Cost considerations severly limited the functions that were designed into the machine and thus limited its usefulness.

The development of the low-cost minicomputer has allowed the newest generation of digital vibration test instrumentation to combine the best aspects of these two approaches into one economical, versatile, high performance system (computers are discussed in the Appendix to this paper). This system can replace most, if not all, of the analog instrumentation in the typical vibration test laboratory.

The basic concept of the computer-controlled closed-loop vibration test system is shown in Fig. 2. The purpose of this paper is to discuss the concepts associated with Fig. 2 with regard to computer-controlled random vibration and reverberation acoustic testing, sinewave vibration testing, and computer-controlled transient (shock) testing. Initially, a brief review of the existing analog systems (Fig. 1) will be stated; throughout the paper, analog techniques will be compared to the digital techniques and advantages or disadvantages of the two techniques will be discussed. Basic concepts will be stressed rather than specific techniques or equipment. General algorithms will be described in the form of block diagrams and flow diagrams. Specific problems and potential problems will be discussed.

II. BASIC CONCEPTS FOR BOTH ANALOG AND COMPUTER-CONTROLLED TEST SYSTEMS

A. Power Amplifiers, Shakers and Conditioning Amplifiers

Figures 1 and 2 both illustrate a power amplifier driving an electrodynamic (or an electrohydraulic) shaker (exciter). The shaker generally has some type of fixture attached to it and, in turn, the fixture holds the test specimen. The forcing function to the shaker is either a random voltage signal, a swept sinewave, or in the case of shock testing, a transient signal. Accelerometers or other transducers such as strain gages respond to the dynamic environment. The response of the test item is indicated only at the points of attachment of the transducers and no assumptions should be made about the test item dynamics except at the transducer points.

The signals generated by the transducers are proportional to acceleration or strain and must be conditioned by special amplifiers. These amplifiers are analog-type devices which produce voltages proportional to acceleration or strain. The gain and sensitivity settings of these amplifiers can be controlled by multiturn potentiometers for the conventional conditioning amplifiers or by the computer in the case of some computer-controlled test systems. In either case, the signals coming out of the conditioning amplifiers are analog signals, and how these signals are processed for test control purposes depends upon whether the test system is of the type illustrated in Fig. 1 or Fig. 2. However, the end result is to control the shaker in such a manner that the test specification is met. Both types of systems sense the response, compare this response to a desired reference, and then correct the output signal to the shaker such that the new response approaches the desired test specification. In the case of transient testing, the test is open loop rather than closed loop. A transfer function is computed between the input to the power amplifier and the response as seen at the output of the conditioning amplifier. The desired transient is mathematically and physically modified by the calculated transfer function in such a manner that when this modified transient is sent to the shaker, the response will, in fact, be the desired pulse in the time domain or the desired shock response spectrum in the frequency domain.

An acoustic reverberation chamber can be substituted for the shaker and a shaped random acoustic spectrum can be controlled on a closed-loop basis using the basic concepts of either Fig. 1 or Fig. 2.

B. Analog Electronic Servos

Figure 1 indicates a sine servo system or random automatic equalizer/analyzer. The sine servo system uses negative feedback to maintain the shaker control signal at the desired operating level. The transducer signal is rectified (Fig. 3) and the resulting DC signal is used to control the amplitude of an independent sinewave generator via a compressor circuit. A compressor speed control circuit provides a means of adjusting the rate at which the servo will correct for changes within the control loop, either automatically or manually. The DC output from the compressor speed control is used to control the amplitude of the sinewave signal passing through the compressor. The independent sinewave generator signal is heterodyned with a high-frequency carrier and single-sideband techniques and/or heterodyne techniques are used to translate the frequency higher in order to facilitate the ease of operating on the sinewave signal. The high-frequency signal then passes through the compressor circuit where its amplitude comes under control of the compressor speed control circuit. The compressor output is then demodulated to provide the lower-frequency servo-controlled test signal output to the power amplifier which in turn drives the shaker to the test specification.

For random excitation control the output of the conditioning amplifier is broken up into narrowband frequency increments by a contiguous series of bandpass filters. The output of these filters is individually rectified and the resulting DC voltage is used to control the gain of individual amplifiers associated with each bandpass filter.

For random excitation control (Fig. 4), the outputs of the conditioning amplifier and the noise generator go to frequency translators, which beat the random noise signals to a higher manageable frequency (as in the case of sinewave control). This higher-frequency spectrum is applied to a bank of contiguous bandpass filters. In the forward loop, the noise generator signal is applied to variable gain amplifiers associated with each bandpass filter. The gain of each of these amplifiers is controlled by the detected signals from an identical set of contiguous filters in the feedback loop. The spectrum shape is initially determined by a set of slide wires associated with each variable-gain amplifier. The output from each variable-gain amplifier is summed and demodulated such that the resulting signal approaches the test specification spectrum.

C. Digital Electronic Servos

It can be seen by comparing Figs. 1 and 2 that portions of the analog and digital test systems are the same. In the analog case, knobs are set by the test operator that will allow the servo to implement the test specification. In the digital system, the test specification must be inserted into the computer in the form of instruction codes. These codes are typically transmitted to the computer by an operator/system interactive conversational language and some minimal knob setting or switch selection. It is in this respect then that the two types of servos conceptually depart in similarity. The purpose of the conversational language program is to obtain, by means of a teletype or other type computer interactive terminal, all the necessary information needed to perform an induced environmental test to any given test specification. These software programs are generally written in assembly language, and it is the means by which the systems operator communicates with the digital control system before, during, and after the environmental test. The conversational language is written as a sequence of subprograms.

The pre-test portion of the program consists of a sequence of questions asked by the computer which must be answered by the systems operator. The questions are asked and answered in a specified format. If a question is answered incorrectly, an error message is generated by the program and the systems operator is allowed to correct his mistake. The pre-test portion of the conversational language program is divided into subcategories:

1. General Test Information. The computer might ask for the test title, number, and date, test engineer, wideband test level, and test duration. The questions are typed out by the computer-controlled teletype or graphic terminal. The systems operator answers each question. The information obtained is tested for format errors, then stored in memory for later use during the actual test. The next question is then formulated by the computer.

2. Frequency Profile. Specifications for an induced vibration environment test are given in the form of a frequency profile versus amplitude. When displayed, the profile can be viewed on a linear-linear, log-linear or log-log coordinate system. For random-type tests, the conversational language program first asks the systems operator for all breakpoint frequencies in any order and then obtains a spectral or slope value for each range of the frequency profile. The acquired information is tested for format consistency and a spectral value is calculated for m spectral lines and n breakpoint frequencies over a bandwidth of p Hertz. Parameters m, n and p are variable. This information becomes the reference spectrum used by the digital system for control. At this time, a printout of the calculated wideband spectrum level is made. For sinewave testing, the program asks for displacement, velocity and/or acceleration levels over the desired frequency ranges. For transient shock testing, the program asks for shock response data information.

3. Time Profile. This section of the conversational language allows the systems operator to specify the startup and shutdown times and how long he wants to control the test at partial and full levels. The systems operator can then choose to obtain a paper tape output and/or a printout of all information and test instructions to this point, prior to the actual test.

The post-test conversational language includes the capability to display, plot, and print out the test results and test analyses in a predetermined format, which completely documents the environmental test. In addition, the identical test can be conducted again either immediately or any time thereafter, since a permanent record of that particular test specification exists on paper tape.

The control program is generally written in assembly language. The program uses the information obtained by the conversational language program to perform all real-time computations needed for system control.

The digital control software determines whether the test specifications are presented via paper tape or from the conversational program. Once the test specification data is obtained by either mode, the control program enters an initialize subroutine. This subroutine initializes the computer by zeroing all the input/output data blocks and sets up all necessary counts; then the system is ready to start the test at the operator's command.

The test specimen response signal is detected by an accelerometer mounted either directly on the test item or on a fixture interfacing the test item to the shaker. The accelerometer signal is conditioned by a transducer charge amplifier. The output of the charge amplifier is a voltage proportional to the test item response acceleration at the mounting point of the accelerometer.

The charge amplifier voltage is the input signal to a computer gain controlled instrumentation differential amplifier. This amplifier is used to isolate the digital system from the shaker system and to allow gain adjustments for full-scale analog inputs to the analog-to-digital converter.

The output of the instrumentation amplifier is coupled to a low-pass anti-aliasing filter whose cutoff frequency is set at least half the sampling frequency of the analog-to-digital (A/D) converter. This satisfies sampling theorem requirements. The filter is used in the Butterworth response mode with a rolloff of sufficiently steep slope.

An analog-to-digital converter samples the output of the low-pass anti-aliasing filter at a rate determined by the test specification. The digitized information is processed by the general-purpose computer and the time series analyzer. The input digitized data is compared to the reference test specification inserted in memory via the conversational language program prior to the actual test. Corrections are made by the computer, and the new forcing function signal in digital form is sent to the digital-to-analog (D/A) converter where it is converted back to an analog signal.

The continuous analog signal from the D/A converter is passed through an output filter whose characteristics and cutoff frequency are identical to the input anti-aliasing filter. The purpose of the output filter is to assure a test frequency spectrum as specified by the test specification and to make the output of the D/A converter a continuous signal.

The filtered analog signal is conditioned by a computer-controlled line driving amplifier whose gain can be changed to produce the required drive voltage signal necessary to drive the vibration system power amplifier.

The power amplifier drives the vibration exciter (shaker) with the modified test specimen response signal such that the new response approaches the desired test specification. The convergence time and control system response are determined by the mechanical dynamics of the test specimen and the algorithm used to determine the error signal. The algorithm constants can be changed by the systems operator.

III. WHY DIGITAL CONTROL OF INDUCED ENVIRONMENTAL TESTS?

The rationale for developing the first random test system was based upon the following ideal advantages[1]:

(1) The test specimen response can be "instantly" displayed on a large cathode-ray tube. Statistical error will be negligible and the confidence level will be high. A quick-look analysis of the response spectrum is essential, particularly as the total test time decreases. For example, the total test time for the Mariner Mars 1969 flight spacecraft was 30 sec. Future random noise vibration tests are expected to be even shorter. Most conventioral control systems can equalize the shaker and test specimen reasonably fast, but cannot adequately display the total response spectrum in a short period.

(2) For complete automation, a preprogrammed test specification can be tape-fed to the computer. Test parameters (levels, test time, etc.) can be verified prior to the actual test. Human error will be eliminated.

(3) The programmed tape input and the use of pseudorandom noise (to be discussed) ensures test repeatability. All computer-controlled environmental laboratories could generate exact test environments. This would be a step toward test standardization.

(4) All analog systems require periodic calibration because of inherent drift. A digital system will minimize these drift and calibration problems. Test results will be more accurate and reliable.

(5) The quick-look readout display capability of a digital system will allow the observation of test specimen irregularities. The computer system will sense the irregularity and abort the test quickly, if necessary. The criteria for test abortion will be preprogrammed into the computer before the actual test.

(6) Protection of the test specimen (test hardware) against transient acceleration and excess steady-state acceleration, velocity, and displacement is of prime importance during all types of vibration testing The digital system can be programmed to monitor these conditions. Should abnormal conditions arise with respect to the programmed "normal" inputs, the computer will abort the test in a controlled manner to protect the test specimen against damage. In addition, the computer could monitor the force/current requirements of the vibration exciter, which in turn will provide test specimen protection against the occurrence of abnormal conditions in the closed-loop control system.

(7) Changes in control philosophy and improvements in testing methods can be easily implemented with appropriate changes to the computer program.

(8) All the important information necessary to obtain a complete test report would be contained within the nondestructive memory of the computer. The test report would be generated after the test by interrogating the computer. Peripheral equipment associated with the digital system would print out the results of the test in a predetermined format, thus eliminating costly computer and analysis time.

(9) The computer control system can also be used for shock pulse and shock spectrum analysis and synthesis. Shock testing with electrodynamic shakers will be more practical with an "on line" computer.

The first random systems were constructed, and these concepts proved to be real advantages. As the sine and shock capabilities were added as features to the computer-controlled test systems, still more advantages over the analog systems were obvious. These advantages were mainly with the fact that the concept of the test system could be changed by changing or adding software in order to meet special-purpose test requirements. These advantages will become obvious in later sections of the paper.

IV. GENERAL FUNCTIONAL REQUIREMENTS FOR COMPUTER-CONTROLLED VIBRATION TEST SYSTEMS

The functions that the test system is to perform can be classified into the following major categories[2]:

(1) Test Signal Generation.

(2) Data Acquisition and Storage.

(3) Data Analysis.

(4) Servo Control of Tests.

(5) Test Documentation.

Each of these categories will be described below.

A. Test Signal Generation

This is a relatively new function that digital systems have taken on. It is now possible to economically generate all the test signals for induced vibration tests. This includes random, sine, and shock signals with the required parameter variations.

1. Random Signal Generation. There are several methods of generating random noise signals using digital techniques. First thoughts on this topic as applied to early proposed computer-controlled test systems for random noise testing are described in Ref. 1, and are repeated here.

To be better able to understand the concepts of random noise generation for computer-controlled test systems, it is necessary to describe some basic electronic components associated with digital noise generation. One such component is the flip-flop or bistable multivibrator.

A flip-flop is a device with two outputs, 0 and 1, mutually exclusive (bistable). The device produces no change in output signal unless a signal (pulse) is applied to its input in accordance with Fig. 5. Assume the 0 and 1 outputs are on and off respectively. A pulse applied to the S or "set" input terminal will turn on the 1 output (1 high) and turn off the 0 output (0 low). A pulse applied to the R or "reset" terminal will turn the 1 output off and the 0 output on. If the 1 output is high and a pulse is applied to the S terminal, or if the 0 output is high and a pulse is applied to the R terminal, no change of output state will occur. If input signals or clock pulses (equally spaced pulses) are applied to the T or "trigger" input of the flip-flop, the output will switch to its other state as shown (Fig. 5).

Another electronic device which can be used for noise generation is the voltage comparator. The voltage comparator is a high-gain, differential-input, single-ended output amplifier. The purpose of this device is to compare a signal voltage on one input with a reference voltage on the other and to produce a logic 1 at the output when the signal voltage is equal to or greater than the reference voltage. When the signal voltage is less than the reference voltage, the comparator output is low (logic 0).

With the flip-flop and the comparator in mind, it is now possible to discuss binary random noise generation.

a. Binary random noise generation. Figure 6 demonstrates conceptually a method of generating what is commonly referred to as binary random noise. A wideband noise voltage from a noise diode is compared with a fixed voltage in such a manner that the output of the comparator triggers the first flip-flop somewhat randomly at a rather high frequency. Both outputs of the first flip-flop go to the S and R inputs of the second flip-flop. In addition, the T input of the second flip-flop is connected to a pulse generator (clock) that is running 20 to 100 times higher in frequency than the highest desired frequency component to be generated. The output of the first flip-flop and the output of the clock are essentially independent events. The output of the second flip-flop is a telegraph wave of several different pulse widths with some DC voltage component. This telegraph wave is applied to a precision switch. The output of the switch is voltage-clamped to ±5 V. The output waveform from the clamp circuit theoretically has zero mean. It is impossible to predict with absolute certainty whether the output waveform at any time is +5 V or -5 V. There is a 50% probability that the binary waveform will change state on any given clock pulse[3]. Clearly, this type of electronic circuit and its output provide an electronic analogy to a coin toss experiment.

The spectrum and amplitude distribution characteristics[2] are illustrated in Fig. 7. The binary random pulses have a $\sin^2 x/x^2$ continuous-frequency spectrum. The spectrum is determined by the constant amplitude (±5 V) of the binary pulses and the clock frequency F_c (or period T), as can be seen in Figure 7a. If the binary pulses are passed through a low-pass filter whose cutoff frequency is approximately 0.45 F_c, the frequency spectrum of Figure 7b can be generated[3]. The slope of the curve above 0.45 F_c is determined by the order of the low-pass filter. The bandwidth of the white noise portion of the spectrum is

[2] A good reference on random noise characteristics is W. Tustin, "Random Vibration Potent Test Tool," EDN Magazine, Cahners Publishing Co., April 1967.

entirely dependent upon the clock frequency. The amplitude distribution of the filtered pulses appears Gaussian, as seen in Fig. 7c[4].

The random noise produced by the method described above is indistinguishable from random noise generated by more conventional techniques, provided the clock frequency is much higher than the highest desired frequency component.

b. Pseudorandom noise generation. To understand the next type of digital noise generation, it is necessary to briefly define two additional electronic components: the shift register and the exclusive-OR gate.

A shift register is formed from a series of flip-flop stages which have been modified so that clock pulses applied to the R input terminals cause a binary pulse (bit) to be shifted through the stages. For example, assume that the first flip-flop of Fig. 8 has been set to represent the logic 1 state (1 output high) and all other flip-flops are in the logic 0 state (0 output high). When a clock pulse is applied to the R or reset line, it will not affect those stages at logic 0, but will clear (reset) the first flip-flop at logic 1. When the first flip-flop is cleared, it produces an output pulse which enters the delay circuit, usually part of the flip-flop. After the clearing pulse duration is terminated, the pulse signal leaves the delay circuit and enters the T or trigger input of the second stage, flipping the second stage to logic 1. A second clock pulse entered on the reset line will clear the second flip-flop (all other flip-flops are in the 0 state), sending a signal into the third flip-flop, which in turn trips that flip-flop to logic 1 after the clock pulse duration has terminated. In short, successive clock pulses progressively shift the binary bit to the right through as many flip-flop stages as are contained in the shift register.

An exclusive-OR gate is a two-terminal input and single-terminal output that gives an output if, and only if, one input is high (logic 1) and the other input is low (logic 0). The exclusive-OR symbol and the logic truth table are shown in Fig. 9.

With the concepts of the shift register and the exclusive-OR gate in mind, it is appropriate to discuss the second, and probably the most important, type of digital noise generation: pseudorandom noise generation.

A shift register and an exclusive-OR gate, if connected as shown in Fig. 10, will generate a special binary random sequence, provided that initially not all the flip-flops are set to logic 0 (0 output high). If this special binary sequence is such that the shift register generates the maximum number of logic 1 and logic 0 combinations possible for n stages of shift register, the binary sequence is called a maximum-length sequence [5]. Although Fig. 10 shows a 4-stage shift register, longer (more stages) shift registers are generally employed and more than one exclusive-OR gate could be used to generate a maximum-length sequence of binary pulses. Where the exclusive-OR gate should be connected to the shift register is not a trivial matter, but tables can be found in the literature covering up to at least 33 stages of shift register [6].

As seen in Fig. 10, the binary pulse train contains 15 bits of all possible logic combinations (except 4 logic 0s) for a 4-stage (n = 4) shift register. In general, with an n-stage shift register, it is possible to generate $2^n - 1$ bits or clock periods before the sequence is repeated, provided the exclusive-OR gate is properly connected. The length of time the sequence lasts before it repeats is determined by $M \cdot (\Delta T)$, where $M = 2^n - 1$ and ΔT is the period of the clock pulse. For example, suppose a 25-stage (n = 25) shift register and a 200-kHz ($\Delta T = 0.5 \cdot 10^{-5}$ s) clock are used to generate a maximum-length sequence. Then, $M = 2^{25} - 1$ or $M = 33,554,431$. The period $M \cdot (\Delta T)$ is equal to 167.77 s or approximately 2.8 min. That is, the binary sequence would exactly repeat itself every 2.8 min. (The sequence has a periodicity of 0.006 Hz.)

The binary pulses from the maximum-length sequence generator appear to be quite similar to the binary pulses generated from the binary random noise generator. In fact, one might say that the binary random pulses are a maximum-length binary sequence of infinite length. However, there is one important difference between the two types of pulse generation: the binary pulses from the maximum-length sequence generator are periodic and deterministic, in contrast to the binary random pulses, which are nonperiodic and probabilistic. Periodic pulses produce line spectra, whereas nonperiodic pulses produce continuous spectra [7]. The line spectra characteristics of maximum-length sequences are illustrated in Fig. 11. Low-pass filtering of the binary sequence produces pseudorandom noise ("random" noise that is periodic). There is no statistical variance associated with pseudorandom noise because it is a periodic deterministic signal. If the spectral lines are very close together, this type of noise has all of the desirable characteristics needed for vibration and acoustic environmental testing, including first-approximation Gaussian-amplitude distribution. But, most important of all, anyone can generate the same binary random chain and therefore the same noise signal, provided the code is known.

Figure 12 illustrates an actual hardware pseudorandom noise generator. The random noise comes from the effects of the low-pass filter. A binary maximum-length sequence exists at TP3. This same sequence can be generated within a computer. That is, a special hardware register need not be built. The sequence can be generated using a software subroutine program and working registers within the control system computer. The results will be number states that will be either +1 or 0, and these states will vary in a pseudorandom manner. Figure 13 is a flow diagram of such a program. These numbers can be used to randomize a complex frequency spectrum to produce phases that yield random noise. In referring to Fig. 13, it can be seen as an example that two 16-bit registers and the overflow single flip-flop C bit register are program-controlled in accordance with the flow diagram. The new C bit (C') is either a 1 or 0, depending upon the previous state of bit 32. The new bit shifted into the first stage of the first 16-bit register is the exclusive-OR of previous bits 32 and 3. As the program is entered, bits 32 and 3 are exclusive-ORed. This results in two possibilities: either a logic 1 or a logic 0. At this time, the computer clock shifts the results of the exclusive-OR output into the first stage, i.e., bit 1. Simultaneously, all the other bits have a 50% probability of changing state. A test is then made on the C bit, and the program can then branch to another subroutine depending upon whether the C bit was a 1 or 0. In other words, the C bit (or any of the other 31 bits) varies in binary state, either 1 or 0, in a pseudorandom manner. The resulting sequence will repeat every four days when clocked at a sufficient rate to yield 2000-Hz bandwidth random noise.

There are many computer algorithms available to generate random numbers which in turn can be used to randomize the phase of a complex frequency spectrum to produce random noise.

As can be seen, then, there is clearly no problem in generating random noise signals using either special hardwired hardware or the control system computer itself.

The philosophical question exists, of course, as to whether central processing time should be allocated to the task of noise generation or whether special hardwired hardware should assume this task independent of the computer.

2. Sinewave Generation. As with random noise, sinewave generation can be accomplished either by the control system computer and a software algorithm or by special dedicated computer-controlled hardware such as a programmable function generator.

a. Computer-generated sinewaves. Induced environmental sinewave testing is almost always accomplished by generating a logarithmic sine sweep as the forcing function. The sweep is generally specified in octaves per minute. The definition of an octave is

$$M = \frac{\ln f_2/f_1}{\ln 2} \text{ octaves} \tag{1}$$

where M is the number of octaves in the frequency range f_1 to f_2 and ln is the natural logarithm operator.

For logarithmic sweeps, the sweep rate is specified in N octaves/min and this is the same as N/60 octaves/sec.

By multiplying both sides of Eq. (1) by ln 2 and then exponentiating both sides, Eq. (1) becomes

$$f_2 = f_1 e^{M \ln 2} \tag{2}$$

Since M has units of octaves, N/60 octaves/sec can be multiplied by units of seconds to yield octaves only as in Eq. (2). That is,

$$M = \frac{N\Delta t}{60} \text{ octaves} \tag{3}$$

where, for logarithmic sine sweeps, $\Delta t = 1/f_1$. So Eq. (2) is of the form

$$f_i = f_{i-1} e^{\lambda \Delta t} \tag{4}$$

where

$\lambda = N \cdot \ln 2/60$

N = sweep rate in octaves/min

$\Delta t = 1/f_{i-1}$

f_{i-1} is the previous full cycle sinewave frequency, i.e., f_1 Hz

f_i is the next full cycle sinewave frequency, i.e., f_2 Hz

To implement Eq. (4) is certainly no strain using the system computer, except that exponentiation takes time and it turns out that some good approximations can make logarithmic sweeps easy and fast to implement. It can be seen that λ is a constant. The function f_i can be approximated by an exponential series; i.e.,

$$f_i = f_{i-1} \left[1 + \lambda\Delta t + \frac{(\lambda\Delta t)^2}{2!} + \frac{(\lambda\Delta t)^3}{3!} + \ldots + \frac{(\lambda\Delta t)^n}{n!} \right] \tag{5}$$

Typically, N is in the order of from 1 to 4 octaves/min over the frequency range from 5 to 2000 Hz. An inspection of the higher-or r terms of the series indicates that these terms add practically nothing to the calculation of f_i. This can be seen in Table 1 for these two sweep rates and frequency limits:

Table 1. Exponential series terms

Frequency (Hz)	Series term	1 octave/min $f_{i-1}\lambda\Delta t$, Hz	4 octaves/min $f_{i-1}\lambda\Delta t$, Hz
5	1	5	5
5	2	1.16 × 10^{-2}	4.62 × 10^{-2}
5	3	1.33 × 10^{-5}	4.27 × 10^{-5}
2000	1	2000	2000
2000	2	1.16 × 10^{-2}	4.62 × 10^{-2}
2000	3	3.34 × 10^{-8}	5.34 × 10^{-7}

(Insignificants of higher-order terms of the exponential series)

Therefore, Eq. (5) is approximated by

$$f_i \cong f_{i-1}(1 + \lambda\Delta t) \qquad (6)$$

But $\Delta t = 1/f_{i-1}$

Therefore,

$$f_i = f_{i-1} + \lambda \quad \text{for upsweeps} \qquad (7)$$

and

$$f_i = f_{i-1} - \lambda \quad \text{for downsweeps} \qquad (8)$$

Equations (7) and (8) then are equations that could be implemented by the control system computer to generate a logarithmically swept sinewave in real-time.

The actual generation can be implemented in the following manner. Assume an allocation of p core locations to store p values of a half sine function (0 to π radians). The sine values can be generated and stored in these allocated core locations by performing the following integer arithmetic

$$\text{TABLE}_i = \text{SIN}\left(\frac{\pi}{p} Q_i\right) \qquad (9)$$

where Q_i = 1, 2, 3, . . . p, and p is the number of points (values) of a half sinewave.

For a 16-bit machine, the sinewave function can run over the numeric integer range from +32767 to -32767 with 0 being at 0°, +32767 at +90°, 0 at 180°, -32767 at 270° and 0 at 360°. For a constant clock rate, the lower frequencies will have more points p than the higher frequencies. The clock rate can be calculated by specifying the minimum number of points p from the allocated core locations to be outputed. That is,

$$\text{CLOCK PERIOD} = \frac{1}{2pf_i} \qquad (10)$$

where p is the number of points to be used to define the half sinewave at frequency f_i.

In order to increase the frequency in accord with Eqs. (7) and (8), it is necessary to output the sine values in the allocated core locations at a faster rate, and if the clock that works the digital-to-analog (D/A) converter operates at a constant rate, this means that fewer values must be put out as the frequency f_i increases. This means that the quality of the sinewave suffers as the frequency increases. However, the fact that the D/A converter output is passed through a low-pass filter guarantees that even a very minimum number of values will yield a low distortion sinewave. As an example, choose a maximum of 4096 values to define a half sinewave and a clock period of 4 µs (250 kHz). From Eq. (10), the low-frequency limit would be about 0.3 Hz, and at 2000 Hz only 62 points could be used to define the half sinewave. Owing to the fact that the function is passed through a low-pass filter, the resulting function would be a clean sinewave of low distortion.

It can be seen from Eq. (10) that a scheme to change the clock period could result in a variable-frequency sinewave that was always defined by the same number of points (values). This concept is difficult to implement with sinewave computer generation techniques but has been implemented with a special hardware sinewave generator [8].

b. Hardware sinewave generators. The first computer-controlled sinewave test system [9] used a minimal amount of core storage since it was strictly sine control. Core storage at that time was costing about $3500 for 4K words (K = 1028) of memory, so the manufacturer chose to design a special computer-controlled function generator to generate the sinewave forcing function, although additional "advantages" were listed (Ref. 8) to provide the rationale for this special hardware. This function generator produced a 128-step synthesized waveform across its frequency range by varying its clock frequency. That is, it employed a variable clock oscillator as the master clock, and the clock frequency was controlled by the control system computer. The variable clock technique was accomplished through implementation of a digitally controlled phase-lock-loop to precisely control the master clock oscillator frequency relative to a highly stable fixed-frequency reference oscillator. The phase lock-loop was commanded by a modulus term while the octave was commanded by an exponent term. These terms are related by

$$f_{out} = K_o A 2^N \qquad (11)$$

where

f_{out} is the output clock frequency

K_o = proportionality constant

A = 0.5 to 1 in 1024 steps

N = 0 to 14 in 15 steps

There are several types of programmable frequency synthesizers suitable for sinewave generation in computer-controlled test systems in addition to the one described in Ref. 8 above.

One type, the direct digital frequency synthesizer, is shown in simplified form in Fig. 14 [10]. The digital-to-analog converter and low-pass filter are the only sections of the system that are not built entirely with digital IC's.

The phase accumulator generates the linearly increasing digital phase value of the sinusoid at a fixed output sample rate. This phase value advances for each output sample by an increment directly proportional to the frequency setting. The read-only-memory (ROM) is a sine function table that converts the phase information provided by the accumulator into digital samples of a sinusoidal waveform.

These samples are fed to the D/A, which produces a staircase approximation of a sinewave in analog form. The low-pass filter removes the higher-frequency sampling components, reducing the waveform to a pure sinusoid.

With reference to Fig. 14, the phase accumulator contains a frequency register. The purpose of the frequency register is to load and store from the control system computer the digital value of the instantaneous desired frequency, as given for example by Eq. (4).

The accumulator consists of an N-bit binary full adder followed by an N-bit data register. At every clock pulse, the adder sums the values in the data register and the frequency register and updates the data register with this new sum, which represents the phase of the sinusoid being generated. Since the accumulator increments at a fixed rate determined by the reference clock, the value in the frequency register is specifically the rate of change of phase.

The accumulator overflows at modulo 2^N, and this cycle of 2^N represents one cycle of the sinusoid. The relationship between N (number of accumulator bits), F_1 (lowest frequency possible), and F_c (CLOCK frequency), is given by

$$F_1 = \frac{F_c}{2^N} \qquad (12)$$

where F_1 represents the resolution of the frequency setting.

The phase samples provided by the accumulator are used to address the ROM in which the sine function is stored. Since the ROM is a relatively expensive device, only the first quadrant values are stored. External circuitry is then used to extend these values for all quadrants.

One version of the accumulator and ROM signal waveforms is illustrated in Fig. 15[11].

The most significant bit (MSB) of the accumulator will be at logical "0" for phase values in quadrants 1 and 2 and logical "1" for values in quadrants 3 and 4. This behavior represents the polarity of the output sinusoid and hence is called the SIGN bit. The next lower bit is logical "0" during quadrants 1 and 3 and logical "1" for quadrants 2 and 4; therefore it is called the QUADRANT bit. These top two bits specify one of four quadrants and are used to extend the first quadrant values of the ROM.

Since there are K address bits into the ROM, the next lower K accumulator bits below the QUADRANT bit are used. The digital value of these bits during one sinusoidal cycle is plotted in Fig. 15. There are four sawtooth cycles, one per quadrant. Using these bits directly for the first quadrant will address the ROM correctly, but for the second quadrant these bits must be complemented so that the slope of the sawtooth is inverted, effectively addressing the ROM from 90° back to 0°. Likewise, for the third quadrant, the address is not complemented and for the fourth it is. The digital value of the address to the ROM is shown in Fig. 15. This waveform represents correct phase addressing for a full-wave rectified or magnitude-only version of one sinusoid cycle. To accomplish this address complementing function, a K-bit parallel complementer circuit is required with complement control provided by the QUADRANT bit.

The digital output of the ROM is shown in Fig. 15, and this corresponds to the expected magnitude-only waveform. The sign or polarity information is contained in the SIGN bit. Since conventional DACs require straight binary or 2's complement digital input codes, this sign-and-magnitude version must be converted. This is accomplished with another complementer circuit which complements the output of the ROM during quadrants 3 and 4. Control of this function is provided by the SIGN bit.

The M bits from the amplitude complementer plus the SIGN bit comprise the M+1 bit word of the complete sinusoid ready for D/A conversion. It is converted and then passed to the low-pass filter. The output of the low-pass filter is a clean sinewave which can be used as the forcing function to the power amplifier. Its amplitude is modified by the control system as required to meet the test specification.

In summary, then, it is possible to either generate the sinewave entirely by the control system computer or use special hardwired hardware controlled by the test system computer to synthesize the sinewave driving signal.

3. Transient Waveform Generation. Transient waveforms are used for computer-controlled shock testing using the test system power amplifier and shaker. These transients are generated by the Fast Fourier transform processor and test system computer. There are several classes of transients currently being used for this type of shock testing [12], some of which are derived from transfer function techniques [13] and described in a later section of this paper

B. Signal Conditioning

Analog signals, generally from multiple sources, must be conditioned, sampled, digitized, and stored in some form of mass memory, such as core, magnetic tape, disc, or multiples of these forms of storage.

Figure 16 illustrates the functions required to convert a single-channel analog signal to digital form prior to entering the control system computer for processing or temporary storage.

1. Analog Signal. The analog signal comes from the transducer signal conditioning amplifier. This analog signal is a voltage proportional to strain or acceleration. For vibration or reverberation acoustic control, this signal's frequency content will usually be below 10 kHz, and in most instances the frequency content of concern is usually below

2500 Hz, since the destructive properties of a forcing function to a structure or component are proportional to the nonrigid body displacements resulting from the forced vibration, and the displacements decrease with increasing frequency by the square inverse frequency. That is, for sinusoidal forcing functions,

$$\text{Peak-to-peak displacement} = \frac{g}{0.0511\, f^2} \qquad (13)$$

where

g = normalized acceleration, and

f = frequency of the forcing function.

At 2000 Hz, a 1-g vibration results only in a theoretical displacement of the test item of about 5 μin. One might worry about these small displacements in the manufacture of semiconductor devices but certainly not spacecraft or aircraft structural members. Therefore, in most instances, the interest of concern bandwidth of the analog signal will be about 2000 Hz, although the analog signal is the result of a response at the mounting point of the transducer, and in most cases this response signal will contain frequencies greater than 2000 Hz.

2. Programmable Amplifier/Attenuator. The amplitude of the analog signal can vary over perhaps a 40 dB (voltage) range in the case of a response signal or open-loop test, in contrast to a control channel. The purpose of the programmable amplifier/attenuator is to modify the analog signal amplitude in such a manner as to guarantee a known maximum (full-scale) input into the remainder of the hardware illustrated in Fig. 16. If the devices illustrated in Fig. 16 are rated at ±10 V maximum full-scale input, it is necessary to guarantee that, at least in the case of random noise testing, the analog signal leaving the programmable amplifier/attenuation be no greater than about 2 V rms. This constraint is necessary in order to guarantee that the 3-sigma noise spikes get digitized and are taken into consideration in determining the rms control level.

For sinewave testing, the two-thirds down from full-scale restriction need not be observed if the control amplitude algorithm is based upon a peak detector concept. If the software detector concept is based upon an average algorithm, then it is necessary that no clipping take place in the data channel.

As can be seen, for large-amplitude signals it may be necessary to attenuate, or in the case of low-amplitude analog signals it may be necessary to amplify, in order to meet the above constraints. The amount of attenuation or amplification is "remembered" by the test system computer and is used as a scale factor in calculating desired as well as actual test levels. The initial setting of the amplifier/attenuator is determined by the system computer and is based upon transducer sensitivity and full-scale test levels specified by the test operator during the conversational mode (inputing the test specification) of the control software program.

3. Programmable Low-Pass (Anti-Aliasing) Filter. The output of the programmable amplifier/attenuator must be passed through a low-pass filter prior to digitizing in order to minimize aliasing errors. The concepts of aliasing are discussed below. Let it suffice for now to state that it is necessary to sample the analog signal from the transducers and conditioning amplifiers at a rate that is at least twice the highest frequency of the analog signals and preferably four times the highest frequency contained within the analog signals. This can be guaranteed by low-passing the analog signals prior to digitizing. For acoustic random, and sine testing, the filter should have Butterworth response; and its cutoff frequency as specified during the conversational mode of the control program should be 0 dB and not the normally accepted half-power point of -3 dB. The conventional transfer function of this type of filter is illustrated in Fig. 17 and the distribution of the poles of the filter are shown in the Fig. 18 S-plane diagram.

The Butterworth-type filter is a "maximally flat" amplitude response filter. This type of filter is generally used for data acquisition systems to prevent aliasing errors in sampled-data applications. The attenuation rate beyond conventional (-3 dB) cutoff frequency is -n6 dB/octave of frequency where n is the order (number of poles) of the filter. For acoustic and vibration control, n should be at least 6, giving an attenuation of at least -36 dB/octave outside the filter passband.

For shock testing, however, the Bessel filter (also known as linear phase) is more suited as the low-pass filter within the data channel of Fig. 16. Figure 19 indicates the transfer function of this type of filter, and Fig. 18 shows the distribution of poles in the complex frequency plane. The Bessel filter is a linear phase filter, and because of this linear phase characteristic, the filter approximates a constant time delay over its passband frequency range. Bessel filters pass transient waveforms with a minimum of distortion and overshoot-to-step inputs. The maximum phase shift is $-n\pi/2$ radians, where n is the order (number of poles) of the filter. The conventional cutoff frequency f_c is defined as the frequency at which the phase shift is one-half of this value. For accurate delay, the cutoff frequency should be at least twice the maximum analog signal frequency. Table 2 lists the conventional -3 dB cutoff frequency as a function of n, the number of poles.

Table 2. Bessel filter cutoff frequency characteristics

	2-pole	4-pole	6-pole	8-pole
-3 dB frequency	$0.77\, f_c$	$0.67\, f_c$	$0.57\, f_c$	$0.50\, f_c$

An "ideal" filter is normally described in the literature as being one with flat response to a cut-off frequency, infinite attenuation beyond that frequency, and linear phase or constant delay over the passband. Such a filter exists only in the minds of mathematicians.

Ideal filters can be approximated by networks that have very large numbers of elements. Alternatively, they can be simulated by lengthy digital computation. But any real filter will have a finite rate of cutoff and will never attenuate at an infinite rate. Also, it will have phase nonlinearities. Therefore, a real filter cannot multiply all of its different frequency components by an equal amount with constant delay. And it can't completely eliminate all the undesired signals.

A real filter, then, will inaccurately multiply the many spectral components of a time-varying signal. The filter's output will differ from the input signal even if its nominal bandwidth is greater than that of the input. To determine the effect of a filter on dynamic accuracy, one must know the spectral distribution of the incoming signal and the transfer function of the filter. Then, by multiplying the original spectrum by the transfer function of the filter, one can derive probable signal accuracy over a period of time for signals having varying amplitude and frequency components.

4. Encoding. The encoding of the analog signal from the filter consists of a sample-and-hold (S/H) device and an analog-to-digital (A/D) converter. For multiple channel control or analyses, a multiplexer would be inserted after the programmable filter in Fig. 16 and would be controlled by the system clock. Each signal channel would have its own programmable amplifier/attenuator and low-pass filter, but would time share the encoder.

Since the encoder takes a finite time to convert the data from analog to digital form, rapidly changing data can cause errors. If the data change during the sampling interval is greater than the value of the least significant bit in the binary word, a spurious output results. The solution is a circuit that samples the analog signal fast, then holds this value as a dc level for the full conversion time. This is the basis of the sample-and-hold circuit.

Figure 20 illustrates the basic idea of a sample and hold circuit. The basic sample-and-hold circuit is a normally open switch that closes for a very short time interval so that the analog voltage value can be stored in a memory element. This value is then held constant, independently of the input signal, while it is encoded. The memory element (often a capacitor) is buffered from the output to minimize decay error.

In the case of several control or analyses signal channels, and where the change in data voltage is more than the least bit during the multiplexer period, the data channel is sampled at the beginning of the multiplexer's normal sample interval. When the multiplexer rise-time is too slow and does not allow enough encoding time, the sample-and-hold circuit takes its analog sample at the end of the multiplexed interval, with the voltage closest to its final value. The encoder then digitizes that voltage while the next data channel is switching to the multiplexer output and is rising to its final accurate value.

The critical step in any analog-to-digital data system is the encoding of analog data. Most encoders use the binary system; their output is the digital equivalent of the analog voltage expressed as a binary number, either as time serial or time parallel pulses or voltage levels. A majority of the existing analog-to-digital encoders are closed-loop analog-to-digital servo systems.

The A/D encoders can be divided into six general classifications (14):

(1) Space encoders.

(2) Chronometric encoders.

(3) Successive-approximation encoders.

(4) Successive-comparison encoders.

(5) Time parallel encoders.

(6) Combination or special technique encoders.

For vibration control systems, successive-approximation encoders are satisfactory. This encoding technique is widely used today and is the fastest conventional time serial logic known. More encoders of this type are probably used in data acquisition systems than any other type.

Starting with the most significant bit, the weight of each binary bit is fed to a voltage summer and the output of this summer is compared against the unknown input voltage. If the summer output is less than the unknown voltage, the bit is left turned on; if the output is greater than the unknown voltage, the bit is turned off. This continues until all of the reference bits have been "stacked" against the unknown voltage. At the end of the series of bit weight tries, a voltage sum is attained which equals the unknown input analog voltage with an accuracy of ±1/2 the least significant bit weight.

Most successive-approximation encoders are closed-loop analog-digital servos consisting of three main elements: a D/A converter (either resistive ladder, capacitive or inductive); an amplifier to detect the difference between input analog voltage and the converter output; and a digital servo.

Resistive and inductive D/A converter ladders are capable of at least 0.01% absolute accuracy and up to 0.001% relative accuracy. Capacitive D/A converters yielded 0.05% accuracies.

The logic of the current summation successive-approximation encoder is identical to that of the voltage summation successive-approximation encoder, but it has the advantage that different input voltage ranges are handled simply by changing the value of the range resistor. Full-scale voltages from 100 mV to 1000 V have been used in actual hardware.

In summary, then, the purpose of the sample-and-hold circuit is to sample the analog signal from the filter and hold the amplitude value of this signal until the A/D converter encodes this analog amplitude voltage to a digital value. Both units are usually controlled by the same clock, but the clock pulse may be delayed in such a manner that the S/H samples the analog signal just prior to digital conversion. Once the conversion has been accomplished, another sample is taken and the process starts over. Naturally, after a conversion, the resulting data word in digital format must be temporarily stored for processing. The rate at which samples of the analog signal are taken is called the sample rate (or sample frequency). This rate, as mentioned above, must be greater than twice the frequency of the cutoff frequency of the low-pass filter, and preferably four times that frequency. For example, if a random test is to be conducted and the test specification requires spectrum control to 2000 Hz, the sampling frequency should be 8000 Hz. This sampling frequency would automatically be selected by the control system computer once the upper breakpoint frequency was specified during the conversational mode of the control program. In addition, the low-pass filter would also be selected for a 0 dB cutoff frequency of 2000 Hz.

5. Sampling Theory. Dynamic continuously varying signals from the filter have a high probability of changing in a manner that is largely unpredictable. The greater the unpredictability over a given period of time, the greater is the amount of information that may potentially be received in that time.

On might think that, if we recover information that determines what was happening all the time, then we must have made observations all the time. This is not true. Any physical system has built-in behavioral limitations that determine how fast its output signal can change. In mechanical systems this is related to inertia. In electrical systems it is related to bandwidth.

It is often thought that the highest sampling speed is needed during the period when the input information is changing most rapidly. But this is not so. When something is moving rapidly, it is likely to continue to move in the same direction. Only when an event ceases does it become impossible to predict what is going to happen next under the influence of random external events.

Consider Fig. 21, which shows a time-varying signal being measured in a time ΔT at time t. Suppose we wish to know the amplitude at time t. The ability to give an accurate answer is based on two factors. One is the ability to determine time t accurately; the other is the ability to accurately measure the amplitude at the precise time t. The true point sample usually referred to in most mathematical analyses cannot be achieved. Rather, there will always be some uncertainty about the exact signal amplitude at the instant of sampling and about the exact time of the measurement. There will always be some jitter in a timing circuit and some finite time required to make a measurement.

Quite clearly, if the signal changes by an amount ΔA during the time ΔT, and if ΔT represents the uncertainty of the time, then there is an uncertainty of ΔA in the single-point sample. The effect of variation of ΔT is not the same for all kinds of analog-to-digital converters. The effect for a successive-approximation converter without sample-and-hold differs from that with sample-and-hold. The effect differs yet again if one uses a ramp converter or a dual-slope converter.

Now, if we make a single-point sample on a time-varying signal, the only information that we obtain is an approximation to the amplitude at that particular time. We do not recover the entire signal. A little intuition reveals that, if we were trying to recover the entire time-varying signal over a period of time, the overall effect of finite measurement time in a series of measurements would be different from that of a single-point sample. For example, if the signal amplitude were continually changing, part of the time it would be increasing upward and part of the time it would be decreasing downward. Analysis of values resulting from a succession of spaced measurements of finite duration will yield more information than a single measurement made at that particular time. This is because the error can be smoothed out.

When an analog signal is digitized it is "quantized." There are only a finite number of possible output codes that any digitizer can yield. An input analog signal may pass through all possible contiguous levels with almost infinite resolution, but the output digital code must "jump" from one value to the next. This is illustrated in Fig. 22, in which it is assumed that when a variable reaches the halfway mark between successive quantization levels the output jumps to the next quantization level. For this condition, assuming an otherwise perfect digitizer, there are points for which the error will be zero. But, most of the time, the resultant code will be either higher or lower than the actual value.

The actual error function for a linearily changing signal is shown at the bottom of Fig. 22. As can be seen, it is a sawtooth signal with a peak amplitude of plus and minus half the quantization level. The rms error for a sawtooth is the peak amplitude divided by $\sqrt{3}$, so the rms quantization error is $Q/(2\sqrt{3})$. Thus when a signal is digitized, "quantization noise" is added to the signal. The relative amplitude of this noise depends on the resolution of the digitizer; its component frequencies depend on the signal form.

If the original signal is later to be recovered, quantization noise must be taken into account. Sometimes one needs to digitize signals that have an extremely wide dynamic range. For example, in a high-fidelity music passage, the dynamic range, from a quiet tinkling sound to a loud boom of percussion instruments, might be 80 dB – or a range of 10,000 to 1. For this example, if the digitizing device did not have a resolution of at least 10,000 to 1, the quiet passages would be missed; or alternatively, the very loud passages would be clipped or distorted.

Figure 23(a) shows the amplitude waveform of a continuously varying signal. If the signal is derived from a real physical system, there are limitations imposed on its maximum rates of change: its first, second, and third derivatives, etc. Because real physical systems have some mass (or equivalent parameter), they are not capable of having such large-amplitude excursions at high rates as at lower rates.

The characteristics of a signal over a period of time can also be described by its spectral distribution, as shown in Fig. 23(b). The Fourier relationships tell us that, if we know the amplitude function of time, we also know the spectral distribution for the signal during that time. Conversely, if we know the spectral distribution and phase relationship between these components for that period of time we also know, uniquely, the amplitude function versus time. A knowledge of one yields a knowledge of the other.

The spectral distribution will normally have a relatively high amplitude at low frequency and lower amplitudes at higher frequency, eventually tailing off to negligible amplitudes. The maximum possible rate of change for a particular parameter may be approximated by summing the maximum rates of change for all possible signal components of different frequencies (assuming that these signal components are in phase).

For random noise, shown as a function of time in Fig. 23(c), the signal may fluctuate continuously. For wideband white noise, the spectral distribution extends over an "infinite" bandwidth, and thus the rate of change of signal level may approach infinity. In real systems, of course, there are physical limitations on the amplitudes of the noise (determined by the noise power) and on the possible rates of change (dependent on actual noise bandwidth).

In Fig. 23(d), the spectral distribution of wideband white noise is shown as being flat. That is, there is an equal probability of energy distribution over each frequency interval. In most unfiltered systems the noise bandwidth, regardless of the level of noise power, will be wider than the signal bandwidth. This means that the noise spectrum will extend out further in frequency, beyond that expected from the actual signal spectrum.

Figure 24 shows a varying signal being sampled in two different ways — at closely spaced points and at not so closely spaced points. If we were to reproduce just the closely spaced points on another sheet of paper, and if the limitations on rate of change of the signal were known, then (without reference to Fig. 24) the points could be interconnected to yield a close approximation to the signal indicated in Fig. 24 by a solid line. If, however, the only points taken were those interconnected by the dotted line, and if just these points were reproduced on another sheet of paper, then an attempt to interconnect the points using the same imposed limitations of rate of change would result in the interconnection shown by the dotted line.

The function indicated by the dotted line appears to be changing at a slower rate than the signal represented by the solid line. The frequency of change implied by the dotted line is called an "alias" of the frequency implied by the solid line.

Thus, if signals are not sampled frequently enough, false information consisting of aliases may result. The low-pass anti-aliasing filter and four times cutoff frequency sample rates reduce this false information (aliasing).

When a signal is sampled (see Fig. 25), the time-varying function $f_r(t)$ can be considered to be multiplied by a sampling function $f_s(t)$. During the sampling time, the sampling function is considered to be of unit height. That is, at that time the varying function is multiplied by a function of zero height. The sampling function $f_s(t)$ then is a series of equally spaced flat-topped pulses having a time duration of ΔT and a repetition rate of $f_s = 1/T_s$. The resultant observed signal, then, will be that shown at the bottom of Fig. 25 at which only regularly spaced portions of the original time-varying function have been observed.

For any particular sampling function, as in Fig. 26, there exists a corresponding spectrum that is precisely calculable. As is well known, flat-topped pulses of constant repetition rate are represented by a series of lines in a spectrum. The line amplitudes are determined by the (sin x)/x function and their spacing is f_s. Thus, there is a dc component in the spectrum of the sampling signal equal to the average level of signal, which is clearly $1 \times \Delta T/T_s$. Lines of lower amplitude occur at f_s, $2f_s$, $3f_s$, $4f_s$, etc.

If it is understood that the original signal is multiplied by the sampling signal having an amplitude of 0 to 1, then a knowledge of transform relationships makes it clear that the original spectrum is being multiplied by the spectrum of the sampling spectrum.

In a digitizing-type sampling system, the resultant multiplied product is not quite as shown in Fig. 25. Only a single number or level is obtained for each sample, not a varying section. This single number obviously contains less information than an analog sample that indicates the variation during the time ΔT. This is because, unlike the digitized sample, the finite-time analog sample is capable of yielding trend or derivative information. Also, the single value obtained by digitizing falsely implies that a true single-point sample in zero time was made. But this is not what takes place physically. It has been shown that the action is approximately equivalent to having passed a single-point sample through a network that spreads the value over the time ΔT. As we know, the network that converts a single-point sample, or impulse function, to a pulse of length ΔT has the transfer function (sin x)/x, where the first null is at frequency $1/\Delta T$.

The effect of not making point samples in zero time thus yields a result similar to having passed the original signal through a low-pass filter with a (sin x)/x characteristic. Therefore, when signals having high bandwidth are to be sampled, one must ensure that the sampling interval is not too long to provide the desired accuracy. It is for this reason that sample-and-hold devices with short aperture times are frequently used prior to digitizing, as in Fig. 16.

We've already seen that if a signal (as shown in Fig. 26) is multiplied by a sampling signal (as shown in Fig. 25), each component of the sampling spectrum will be multiplied by the original spectrum The original spectrum will be multiplied by a dc component, thus yielding itself; and, in addition, it will be multiplied by f_s to produce an additional spectrum symmetrically around f_s. This occurs because, when two signals having different frequencies are multiplied, the product contains all the sums and differences of the two frequencies. Similarly, there will be symmetrical spectra around the harmonic frequencies $2f_s$, $3f_s$, $4f_s$, etc.

If the original spectrum is characterized by that of an "ideal" filter, then, as shown in Fig. 27, the spectra from sampling multiplication will be rectangular. The first one, starting at zero frequency, has a width of f_{co}, while the other spectra surrounding $2f_s$, $3f_s$, etc., will each have a width of $2\ f_{co}$.

After sampling, then, the spectrum of the resulting signal is as shown in Fig. 27. If, however, we wish to recover the original signal, we must eliminate the spectra centered around f_s, $2f_s$, etc. This is achieved by filtering out frequencies other than the low-band spectrum which represents the original signal.

Thus, if an ideal low-pass filter were employed which passed the low-band spectrum but did not pass the spectra surrounding f_s, $2f_s$, etc., the output would be the original signal, unmodified.

Suppose that our ideal low-pass filter had a bandwidth of just f_{co} and thus was perfectly flat over the range from zero to f_{co}, attenuating all other frequencies at an infinite rate. It would not modify the original low-band spectrum, and it would eliminate all components of the sampling spectra around f_s, $2f_s$, etc.

For this unrealistic supposition, we can immediately see the lowest mathematically possible adequate frequency of sampling. We can move f_s downward in frequency until $f_s - f_{co}$ just equals f_{co}. Then, solving for f_{co}, it is quite clear that $2f_{co} = f_s$. If f_s were made lower than $2f_{co}$, the low-frequency end of the spectrum surrounding f_s, which contains the high-frequency end of the original spectrum, would "fold over" into the original spectrum and the folded signals could not be separated from or distinguished from the original signal. Thus, $f_s/2$ is considered to be the lowest frequency at which a signal having frequency components up to f_{co} can be sampled and still theoretically be recovered without inaccuracies. This is the theorem yielding Nyquist's result; i.e., equispaced data, with two or more points per cycle of highest frequency, allows reconstruction of band-limited functions.

However, real signals and their spectra are not characterized by ideal low-pass functions; nor are ideal filters available for recovery separation as previously mentioned.

Figure 28 represents more realistically what may happen. The original signal spectrum tends to have a decreasing amount of energy at upper frequencies. Superimposed on the signal spectrum will probably be a noise spectrum that tails off at a much slower rate. When the combined signal-plus-noise spectrum is multiplied by a sampling signal rate f_s, there results a new spectrum which includes the original spectrum plus spectra symmetrically located about f_s, $2f_s$, etc.

Thus, for a real signal characterized by a spectrum decreasing at a finite rate and never reaching zero amplitude, there is no real value of f_s equal to twice the highest frequency present. Further, there is no real filter available that can cut off at an infinite rate and separate the spectra.

Therefore, to determine the minimum allowable sampling rate in a practical situation we must carefully analyze the original spectrum and its noise content. And we must use our judgment to determine what degree of error and distortion can be tolerated. To decide on a lower limit for f_s, the test engineer must know the quality of the filter that can be used to separate the original spectrum from the higher-order spectra. He must make a judgment not as to whether there will be any signal folded into the original spectrum but as to how much he can tolerate – and not whether he can allow any noise to be folded into the original spectrum, but how much he can tolerate.

The recovered signal must be obtained with a minimum time delay so that the sampled data can be used in a real-time control loop. This imposes further restrictions on the nature of the recovery filter. As mentioned earlier, the more complex a physical filter the greater will be its time delay. (A similar argument applies to digital computation.)

Thus, quite clearly, a sampling system must be considered to be just that: a system. The overall transfer function must be considered if any realistic estimate of the accuracies to be achieved are to be obtained. The test engineer must take into account the noise level, the noise bandwidth, and the shape of the original spectrum. He must determine what amount of frequency folding he can tolerate and, then, by considering all of the factors combined, he can determine what minimum rate of sampling to use.

It has been determined experimentally from recent computer controlled test systems that the digitizing process is adequate when the low-pass antialiasing filter has at least 8 poles (-48 dB/octave, Butterworth response), the A/D converter is at least a 12-bit device, and the sampling frequency is four times the 0-dB cutoff frequency of the filter. As mentioned above, one truly never gets rid of the aliasing errors but only reduces them to a level not worth worrying about for the control application.

6. Digital-to-Analog Conversion. Digital-to-analog (D/A) conversion is necessary in the computer controlled test system in order to implement the following important functions.

(1) Produce an analog drive signal to the power amplifier.

(2) Produce analog signals to the oscilloscope display devices.

(3) Produce drive signals to the X-Y recorder.

Item 1 above necessarily includes, besides a good quality D/A converter, a programmable amplifier/attenuator and low-pass filter. The filter is normally identical to the input anti-aliasing filter and is necessary to reconstruct a true analog signal from the D/A converter. Outputs from unfiltered D/A converters exhibit signals that are step voltages and appear as synthesized waveforms. The filter removes the discrete steps or points by eliminating the high frequencies associated with discrete steps. Figure 29 is a block diagram of the analog output requirements of a computer-controlled induced environmental test system. The programmable amplifier/attenuator determines the amplitude voltage requirement to the power amplifier in order to meet the test specification across the desired frequency spectrum. The actual value of attenuation or amplification determines the loop gain as does the input amplifier/attenuator. These gains are determined by the system during the conversational language mode of the control program and are dependent upon transducer sensitivity, internal scale factors, and test level.

7. Trigger Logic. The trigger logic circuit (Fig. 16) is necessary for the analysis function of the computer-controlled test system and is used for shock pulse analysis. The purpose of the trigger logic is to start the analog-to-digital encoding when the test system senses the presence of the leading edge of an analog shock pulse signal. This provides an automated means of starting the data conversion a split second after detection of the shock pulse. This guarantees that the shock pulse will be captured by the test system.

In some systems it is possible to trigger on either a positive or negative slope and, in addition, the operator has a choice as to where (in amplitude) on the slope of the leading edge to start the analysis.

8. Signal Conditioning Errors. Table 3 lists error factors and potential error factor problems associated with the data acquisition and conditioning system illustrated in Fig. 16. This acquisition and conditioning system is also necessary for the third function that the computer-controlled test system serves to the environmental laboratory, which is described next.

Table 3. Error factors from input data channel

Analog signal

- Transducer nonlinearities.

Programmable amplifier/attenuator

- Input impedance.
- Input current.
- Offset.
- Transfer accuracy.
- Linearity.
- Interference susceptibility.

Programmable low-pass (anti-aliasing) filter

- Amplitude response.
- Phase response.
- Transient response.
- Linearity.
- Transfer accuracy.

Multiplexer (for multiple-channel control/analysis)

- Cross talk.
- Transfer accuracy.
- Leakage current.

Sample-and-hold

- Transfer accuracy.
- Droop.
- Offset.
- Aperture time.
- Acquisition time.

Analog-to-digital converter

- Speed.
- Rate.
- Mode of conversion.
- Resolution.
- Monotonicity.
- Stability.
- Interference susceptibility.

C. Signal Analysis

The third function that the computer-controlled test system provides for the Induced Environmental Test Laboratory is the time series analysis function, and in particular, the following frequency functions

(1) The Fourier transform.

(2) The power spectral density function $G_x(f)$.

(3) The cross spectral density function $G_{xy}(f)$.

(4) The frequency-response function H(f).

Some or all of these functions are required for computer-controlled random vibration tests and transient shock tests.

1. Windowing. In digital time series analyzers, it is necessary to window the incoming data from the A/D converter prior to calculating some of the above frequency functions. This is due to the spectral content of the sampling function f_s, shown in Fig. 26, and, in fact, the ill effects of this spectrum past the frequency $1/\Delta T$. The sample function f_s is also referred to as the window function (in the frequency domain) or the "boxcar" function in the time domain(16). The raw estimate of the above-listed frequency functions is the convolution of the true function with the sample function in the frequency domain or their products in the time domain. The effect of the sample function causes errors in the estimated spectral values of these frequency functions. To reduce these errors, the digital analyzer portion of the control system must employ data weighting. Weighting or windowing such as the Hanning function, described below and in Fig. 30, smooths the raw data estimates by modifying the normal "boxcar" time domain function and its associated (sin x)/x type frequency function. Windowed analyses result in better frequency function estimates than non-windowed analyses. Many types of weighting functions are defined, and the choice ultimately is a factor in the upper bound of the control systems real-time processing capability. The Hanning window or weighting can be accomplished in the time domain or in the frequency domain after the spectral values of the frequency functions have been generated. For the latter case, assume an unHanned data block of spectral values stored in BLK1 of elements x(k), k = 1, 2, ..., n. After the Hanning weighting, define another data block of unnormalized weighted spectral values to be stored in BLK2 of elements y(k), k = 1, 2, ..., n. Then

$$y_1 = \frac{1}{2}(x_1 + x_2) \tag{14}$$

$$y_2 = \frac{1}{4}x_1 + \frac{1}{2}x_2 + \frac{1}{4}x_3 \tag{15}$$

$$\vdots$$

$$y_i = \frac{1}{4}x_{i-1} + \frac{1}{2}x_i + \frac{1}{4}x_{i+1} \tag{16}$$

$$\vdots$$

$$y_{n-1} = \frac{1}{4}x_{n-2} + \frac{1}{2}x_{n-1} + \frac{1}{4}x_n \tag{17}$$

$$y_n = \frac{1}{2}(x_{n-1} + x_n) \tag{18}$$

A general algorithm for performing this process is illustrated in Fig. 30. The above Hanned values are "unnormalized" and must be multiplied by a factor of $(8/3)^{1/2}$, assuming the function being Hanned is a Fourier transform and the x(k)'s are Fourier coefficients. If the x(k)'s are spectral values of the PSD or cross spectrum function, then the multiplying factor is 8/3. This will become apparent when the PSD function is described in this section of the paper. It turns out that it is most desirable to window the Fourier coefficients provided these coefficients are used to generate functions of frequency and not the transforms (corresponding time functions) of these frequency functions.

The windowing or weighting described above is a process that must be done on the input data from the A/D converter, or immediately after the Fourier coefficients have been generated in order to control a random noise vibration or acoustic test. There is another type of so called "windowing" which is performed on the output signal to the power amplifier. This windowing is used to concatenate output frames together in the random vibration control process.

2. The Fourier Transform.[3] The Fourier transform is the basic frequency function that a computer-controlled test system or a time series analyzer performs. It enables the system to compute all the other time and frequency functions such as correlations, density functions, transfer functions, and the coherence function.

The Fourier transform in essence is a mathematical operation performed on a time history function to determine and analyze its frequency content. The results of the Fourier transform process are therefore the magnitudes and the phases, called complex Fourier coefficients, that the signal possesses at the various frequencies within the control (or analysis) bandwidth. In digital systems, the input time signal is digitized as mentioned previously so that it is represented by a finite number N, called the frame size (FS), of amplitudes. The Fourier transform operation performed is therefore necessarily a discrete rather than a continuous process. The general form of the complex Fourier coefficients is:

$$a_i = \sum_{k=0}^{N-1} x_k W^{-ik} \tag{19}$$

$$i = 0, 1, 2, \dots, N-1 \; (W = e^{-2\pi j/N}, \; J^2 = -1)$$

where N is the frame size and x_k is the kth digitized time sample. This method of calculating the Fourier coefficients involves N complex operations (i.e., complex additions and complex multiplications) for each coefficient, i.e., N^2 complex operations in all.

[3] Subsection 2 was written by W. Boctor.

Considerable savings in time can be achieved by reducing this number from N^2 to $N \log_2 N$. This is realized by performing the Fourier transform in steps, following the Cooley-Tukey(17) algorithm. If m steps are involved, then the number of operations become $N (r_1 + r_2 + \ldots + r_m)$, such that $(r_1 \cdot r_2 \cdot \ldots \cdot r_m) = N$, and r_i ($i = 1, 2, \ldots, m$) are the radixes used in the different steps. Although the minimum number of operations is achieved by choosing all $r_i = 3$, $r_i = 2$ proves to be much more advantageous for use by digital computers because of their binary nature. The loss in speed over $r_i = 3$ is only about 6%. This yields what is now known as the radix-2 algorithm and the Fourier transform thus calculated is called the fast Fourier transform (FFT). This method provides the relationship

$$N = 2^m$$

and number of operations = $N \log_2 N$.

This sets the imperative requirement that FS be an integral power of 2. A flowchart that tests for this relationship is shown in Fig. 31.

In this algorithm, the results obtained in the (n-1)th stage are utilized in calculating 2^{m-n} sums of the form

$$s_{i,n} = \sum_{k=0}^{2^n-1} s_{k,n-1} W_n^{-ik}$$

$$i = 0, 1, 2, \ldots, 2^{n-1}$$

$$(W_n = e^{2\pi j/2n}) \tag{20}$$

This goes on until all m stages are performed, the results of each stage depending upon and spacially replacing the results of the previous stage. The FFT thus calculated is called an in-place FFT because the output block containing the real and imaginary parts of the Fourier coefficients is coincident with the input block which previously contained the input time signal. In this case of a real input, the output block is skew-symmetric about the dc value, and only the positive frequency components need be calculated and retained. Thus an N-size real-time input block yields N/2 real frequency components and N/2 imaginary frequency components, and the FFT can still be performed in place. However, this requires the input time block to undergo first a shuffling process so that no components within it get replaced in any stage by others (resulting from computations in that stage), except when they are not needed any further. Fig. 32 shows an example for a flowchart of the shuffling process.

An example of the main FFT flowchart is given in Fig. 33. It is seen that the main algorithm is used for the inverse Fourier transform (IFT) also, which is used to compute the correlation functions from the spectral density functions in analysis, and more important, it is used in control to compute the signal to be sent out to the shaker or other exciter from the drive spectrum. The factor ISIGN is -1 for an FFT and +1 for an IFT.

It is worth noting that the cosines and sines used in the algorithm are generated in non-real-time and saved as a lookup table, and depending upon the frame size N, only every (N_{max}/N)th valve is used, where N_{max} is the maximum system frame size. Usually sines and cosines of angles within the range 0 to 90 deg only are generated because of the large memory storage required (in software FFT) or the large ROM (read only memory) size required (in hardware FFT). Mathematically,

$$\cos(\text{ISIGN}, M, L) = \cos(\text{ISIGN} * 2\pi * (M-1)/L) \tag{21}$$

$$\sin(\text{ISIGN}, M, L) = \sin(\text{ISIGN} * 2\pi * (M-1)/L) \tag{22}$$

The sine-cosine table, whether implemented in software or hardware, is usually in integer form to speed up the multiplication operations and is such that a sine or cosine value of 0 is represented by 0 whereas a sine or cosine value of 1 is represented by the largest integer number the computer registers can accommodate (32767 in a 16-bit machine). This offers two great advantages: (1) it achieves the highest resolution possible with the existing machine, and (2) when multiplying any number (in the multiplicand register) by a value from the table after moving it to the multiplier register (again whether software or hardware), the product obtained in both registers may be chopped off with the original register (the multiplicand register) containing the correct 16-bit value of the product. No further shifting or normalization is required. The reason here is that with the sine-cosine table in this form, all its values are inherently multiplied by 2^{n-1}, where n is the register size in bits (i.e., multiplied by 32767 in this example). And chopping off the right half of the product is equivalent to a division by that same factor, so that the end result in the multiplicand register is the correct one.

The FFT/IFT algorithm described above is suitable for both software and hardware implementation. The former is less expensive than the latter but it is considerably slower (about 80 to 100 times on the average, as shown in Table 4). Hardware FFT/IFT is therefore more suitable in digitally controlled testing. In this case the sine-cosine table is a read-only-memory containing all the sine and cosine values within the first quadrant for fast access. The complex operations are all performed in integer form in hardware registers. The fast Fourier processor, as it is called, transfers the input block from the computer main storage to a

Table 4. Comparison of execution times (in ms) of forward and inverse Fourier transforms in both software and hardware implementation (hardware data are according to highest speeds commercially available in January 1974).

Frame size	Execution time, ms			
	Software		Hardware	
	FFT	IFT	FFT	IFT
64	32	38	0.58	0.58
128	75	90	1.09	1.09
256	170	200	2.07	2.07
512	380	450	4.9	4.9
1024	850	1000	9.8	9.8
2048	1650	2100	21.7	21.7
4096	3200	4300	37.8	37.8

buffer block within it, performs the FFT or IFT on it, and puts the result back in the original main storage block in the computer memory.

3. The Power Spectral Density Function. The power spectral density function (PSD) $G_x(f)$ is also known as the normalized power spectrum, the normalized auto spectrum, and the auto spectral density function. In this paper, the function $G_x(f)$ will be called the power spectral density function or simply PSD function.

The $G_x(f)$ is defined as the self-conjugate product of the real and imaginary coefficients of the Fourier transform of the time function x(t). That is,

$$G_x(f_i) = X(f_i)\, X^*(f_i) \qquad (23)$$

where

$$X(f_i) = a_i + jb_i \qquad (24)$$

and * means complex conjugate. So

$$G_x(f_i) = (a_i + jb_i)(a_i - jb_i) \qquad (25)$$

giving

$$G_x(f_i) = a_i^2 + b_i^2 \text{ at frequency } f_i \qquad (26)$$

The coefficients a and b are the real and imaginary coefficients at the frequency f_i of the Fourier transform of x(t). The j operator drops out in the multiplication process ($j^2 = -1$). What this means physically is that there is no phase information in the PSD function.

For vibration analysis and control, it should be understood that the Fourier coefficients, as they are generated within the system processor, have units of peak voltage, and since the coefficients are derived from sine and cosine tables, these coefficients should be multiplied by a scale factor of $1/\sqrt{2}$ in order to obtain rms amplitudes.

If the scale factor is applied after $G_x(f_i)$ is generated, then the scale factor becomes 0.5 times $G_x(f_i)$. In addition, if Hanning weighting is accomplished after $G_x(f_i)$ is generated, the overall scale factor becomes 1.33.

If the signal being analyzed is random in nature, the PSD function can be normalized by dividing the results by the effective filter bandwidth used in the analysis to provide a characterization in terms of 1-Hz bandwidth. This allows direct comparisons of analyses to be made. The PSD, as defined here, is the normalized function that has rms units of g^2/Hz. The PSD of a random variable is an estimate in the strictest sense, and the term PSD in this paper always implies an estimate.

Figure 34 illustrates the basic process required to implement the PSD function using a digital machine designed around fast Fourier transform concepts. A PSD analysis is dependent upon the following parameters associated with all digital analyzers.

(1) Frame size (FS) and spectral lines (SL): A Fourier analysis requires n real values in the time domain in order to generate n/2 spectral values (spectral lines) in the frequency domain. The frame size is the number of data words in the time domain to be processed at one time. The spectral lines resulting in a PSD analysis is one-half the frame size number selected for the analysis.

(2) Sample rate (SR), Sample period (SP): The sample rate is the points per second selected by the operator for the analog-to-digital conversion associated with digital analyzers. The units of sample rate are points per second, samples per second, or conversions per second. The sample period is the reciprocal of the sample rate.

(3) Mode (M): The mode, continuous (real-time) or discontinuous (non-real-time), is a function of the sample period. If the sample period SP is too short (i.e., the sample rate too fast) for the number of spectral lines selected, the analyzer will indicate and automatically go into a discontinuous mode of operation.

(4) Analysis bandwidth (AB): The analysis bandwidth is defined as some fraction of the sample rate; the units are hertz. The analysis bandwidth must satisfy the minimum requirements of the sampling theorem

$$AB < 0.5\ SR\ Hz \tag{27}$$

or

$$AB < \frac{1}{2SP}\ Hz \tag{28}$$

(5) Effective filter bandwidth (FB): The effective filter bandwidth is defined as the analysis bandwidth divided by the spectral lines

$$FB = AB/SL = SR/FS\ Hz \tag{29}$$

(6) Load time (LT): The load time is defined as the time required by the digital analyzer to obtain one frame of data commensurate with the selected frame size. The load time is the indicated frame size divided by the sample rate

$$LT = FS/SR \tag{30}$$

or

$$LT = (FS)\,(SP) \tag{31}$$

The load time, digitally speaking, is equivalent to averaging time in the analog domain. The effective filter bandwidth and load time product is always one.

$$(FB) \cdot (LT) = 1 \tag{32}$$

(7) Analysis time (AT): The analysis time is the actual time required to calculate a total estimate of K statistical degrees of freedom from time frames of input data.

$$AT = K \cdot LT/2 \tag{33}$$

(8) Statistical degrees of freedom (K): The number of degrees of freedom of a spectral estimate is a convenient description of the statistical reliability. The number of degrees of freedom K per frame is defined as twice the effective filter bandwidth times the load time.

$$K/frame = 2\,(FB)\,(LT) = 2 \tag{34}$$

For all digital analyzers, a single spectral estimate of two degrees of freedom is produced from each time frame of Gaussian distributed data since the effective filter bandwidth and the load time product is always 1.

As more time frames are analyzed, the degrees of freedom go up, indicating better reliability of the spectral estimate. It is important, therefore, to express a quality factor of degrees of freedom per second of analysis time.

$$K/sec = 2/LT \text{ for continuous mode}$$

and

$$K/sec = 2/(LT + FS \cdot SP) \text{ for discontinuous mode}$$

The above parameters are all in effect during a PSD analysis. Some of the parameters must be selected prior to the process implementation illustrated in Fig. 34. These parameters are generally the analysis bandwidth and effective filter bandwidth as well as the number of averages required in order to obtain acceptable statistical reliability. The selection of these parameters sets the remaining parameters such as frame size and sample rate. Load time is a physical result from the previous choices as is the statistical reliability once the number of averages have been selected. The analysis time is a function of the algorithm, hardware and software. The parameters are selected or calculated during the conversational mode of the control program.

The area under the PSD locus is the mean-square value of x(t); the rms of x(t) is the square root of the mean-square value. The digital analyzer must perform a numerical integration. This can be accomplished by implementing the trapezoidal rule or Simpson's rule. For vibration and acoustic analysis the trapezoidal method of integration is generally adequate and easy to implement by dividing the first and last spectral densities by 2 and adding these values to the sum of all the other spectral densities. The final sum is then multiplied by the effective filter bandwidth to obtain the mean-square value of x(t). A square root algorithm is then required to obtain the rms value of x(t).

4. The Cross Spectral Density Function. The cross spectrum function $G_{xy}(f)$ is also called the cross power spectrum or, when normalized to 1 Hz, cross spectral density function. The cross spectrum function relates to two time signals, both of which could appear as either forcing functions or response functions but usually an input forcing function x(t) and a response function y(t). In any case, the Fourier transforms X(f) and Y(f) are required to obtain $G_{xy}(f)$, where, at frequency f_i,

$$X(f_i) = a_i + jb_i \tag{35}$$

and

$$Y(f_i) = c_i + jd_i \tag{36}$$

A cross-conjugate product is performed in the computer control system as follows:[4]

$$G_{xy}(f_i) = (a_i - jb_i)(c_i + jd_i) \qquad (37)$$

or

$$G_{xy}(f_i) = a_ic_i + b_id_i + j(a_id_i - b_ic_i) \qquad (38)$$

where a_i, b_i, c_i, and d_i are the Fourier coefficients at frequency f_i of the time functions x(t) and y(t), respectively.

It can be seen from Eq. (38) that the cross spectrum is a complex frequency function; that is, it has a real and imaginary part, unlike the power spectral density function $G_x(f)$. Let

$$P_i = a_ic_i + b_id_i \qquad (39)$$

and

$$Q_i = a_id_i - b_ic_i \qquad (40)$$

Then

$$G_{xy}(f_i) = P_i + jQ_i \qquad (41)$$

where P is called the coincident term or simply co term and Q is called the quadrature term or quad term. For rms amplitude units, Eq. (7) must be multiplied by a scale factor of 0.5, and a scale factor of 8/3 for Hanning weighting, for a total scale factor of 1.$\overline{33}$.

Equation (41) is the rectangular form of the cross spectrum function. The function should be normalized to 1 Hz by dividing by the effective filter bandwidth. The normalized polar form is

$$G_{xy}(f_i) = \left(P_i^2 + Q_i^2\right)^{1/2} \; g^2/\text{Hz} \;\; \angle\theta_{xy}(f_i) \qquad (42)$$

where

$$\angle\theta_{xy}(f_i) = \text{TAN}^{-1}\, Q_i/P_i \text{ rad} \qquad (43)$$

It can be seen that the cross spectrum function $G_{xy}(f)$ yields both amplitude and phase information.

From Eqs. (39), (40) and (42),

$$\text{MAG } |G_{xy}(f_i)| = \left[\left(a_ic_i + b_id_i\right)^2 + \left(a_id_i - b_ic_i\right)^2\right]^{1/2} \qquad (44)$$

or

$$\text{MAG } |G_{xy}(f_i)| = \left[\left(a_ic_i\right)^2 + 2a_ib_ic_id_i + \left(b_id_i\right)^2 + \left(a_id_i\right)^2 - 2a_ib_ic_id_i + \left(b_ic_i\right)^2\right]^{1/2} \qquad (45)$$

rearranging and cancelling some terms,

$$\text{MAG } |G_{xy}(f_i)| = \left[a_i^2\left(c_i^2 + d_i^2\right) + b_i^2\left(c_i^2 + d_i^2\right)\right]^{1/2} \qquad (46)$$

or

$$\text{MAG } |G_{xy}(f_i)| = \left[\left(a_i^2 + b_i^2\right)\left(c_i^2 + d_i^2\right)\right]^{1/2} \qquad (47)$$

From the definition of the power spectral density function (PSD),

$$\text{MAG } |G_{xy}(f_i)| = \left[G_x(f_i)\, G_y(f_i)\right]^{1/2} \qquad (48)$$

Therefore, either Eq. (42) or Eq. (48) can be used to calculate the normalized magnitude of the cross spectrum function from the digital analyzer.

[4] $G_{xy}(f_i)$ could be calculated by the cross conjugate product of $(a_i+jb_i)(c_i-jd_i)$. However, this product will result in a phase angle 180 deg apart from the frequency response function H(f), defined next in this paper.

The cross spectrum function by itself is not widely used. It is used to calculate the frequency response function described next in this paper.

The cross spectrum function and the cross correlation function are transform pairs in the frequency and time domain, respectively. The power spectral density functions $G_x(f)$ (input forcing function), $G_y(f)$ (output response) of x(t) and y(t), respectively, and the cross spectrum of x(t), y(t), i.e., $G_{xy}(f)$, will yield the total frequency information required for an understanding of the physical process resulting from the excitation of a structure, provided that the cross spectrum is measured at both transducer points simultaneously and both PSD functions are derived from that same data.

Figure 35 shows the processes required to obtain the cross spectrum function. All of the parameters defined for the PSD function are in effect for the cross spectrum function. These parameters include:

(1) Frame size or number of spectral lines desired for the analysis.

(2) Sample rate or sample period.

(3) Analysis bandwidth.

(4) Effective filter bandwidth.

(5) Load time.

(6) Analysis time.

(7) Statistical degrees of freedom.

For strictly analysis purposes, a two-channel analyzer is required with dual analog-to-digital converters. These converters must share a common clock such that the conversions associated with each converter are simultaneous. Sets of Fourier coefficients are generated from the Fourier transforms of x(t), the input forcing function, and y(t), the response function. A cross-conjugate multiply process is performed on these Fourier coefficients at each center frequency i of the effective filter bandwidth across the desired analysis bandwidth. For control purposes, it may be desirable to take the cross spectrum of certain signals associated with the control process. This will be discussed in a later section of this paper.

5. The Frequency Response Function. Consider Fig. 36A. The reason x(t) is not identical to y(t) is that the test item itself is characterized physically by its impulse response function h(t).[5] The impulse response of the item under test can be considered as a mathematical model for the test item in the time domain, since it can be used to relate the input x(t) and the output y(t). In the frequency domain, the impulse response function h(t) transforms to the frequency response function H(f), as seen in Fig. 36B. That is, the Fourier transform of h(t) is H(f). The frequency response function H(f) goes by several names. It is also called the transfer function and the system function, but strictly speaking, H(f) is the frequency response function, and the term transfer function should be reserved for the Laplace transform of the impulse function h(t), i.e., H(s).

The frequency response function at frequency f_i is defined as

$$H(f_i) = \frac{Y(f_i)}{X(f_i)} \tag{49}$$

where $X(f_i)$ and $Y(f_i)$ are the Fourier transforms at frequency f_i of x(t) and y(t), respectively, in Fig. 36B, and

$$X(f_i) = a_i + jb_i \tag{50}$$

and

$$Y(f_i) = c_i + jd_i \tag{51}$$

where a_i, b_i, c_i and d_i are the real and imaginary coefficients of the Fourier transforms at frequency f_i, and j is the imaginary operator $\sqrt{-1}$.

Therefore,

$$H(f_i) = \frac{c_i + jd_i}{a_i + jb_i} \tag{52}$$

In polar form, Eq. (52) becomes

$$|H(f_i)| = \frac{C_i}{A_i} \angle \underline{\theta_{xy}(f_i)} \tag{53}$$

where

$$C_i = \left(c_i^2 + d_i^2\right)^{1/2} \tag{54}$$

and

$$A_i = \left(a_i^2 + b_i^2\right)^{1/2} \tag{55}$$

The phase angle

$$\angle \underline{\theta_{xy}(f_i)}$$

[5] h(t) is also known as the weighting function of a constant parameter linear system.

can be calculated by multiplying Eq. (52) by the complex conjugate of $a_i + jb_i$, i.e., $a_i - jb_i$ to obtain

$$\underline{/\theta_{xy}(f_i)} = \mathrm{TAN}^{-1} \frac{a_i d_i - b_i c_i}{a_i c_i + b_i d_i} \text{ rad} \qquad (56)$$

This phase angle is identical to the phase angle of the cross spectrum function at all frequencies across the analysis bandwidth of the frequency response function.

Equations (54) and (55) are the square roots of the power spectral density functions $G_y(f_i)$ and $G_x(f_i)$, respectively (refer to the power spectral density function description in this paper). That is,

$$C_i = \left(G_y(f_i)\right)^{1/2} \qquad (57)$$

and

$$A_i = \left(G_x(f_i)\right)^{1/2} \qquad (58)$$

Therefore,

$$|H(f_i)| = \left(\frac{G_y(f_i)}{G_x(f_i)}\right)^{1/2} \underline{/\theta_{xy}(f_i)} \qquad (59)$$

From Eq. (48), it is noted that

$$\mathrm{MAG}|G_{xy}(f_i)| = \left[G_x(f_i)\, G_y(f_i)\right]^{1/2} \qquad (60)$$

Dividing both sides of Eq. (60) by the PSD of the input function x(t), i.e., $G_x(f_i)$, the results are

$$\frac{\mathrm{MAG}|G_{xy}(f_i)|}{G_x(f_i)} = \left(\frac{G_y(f_i)}{\ _x(f_i)}\right)^{1/2} \qquad (61)$$

The right-hand side of Eq. (61) is precisely the magnitude of the frequency response function (Eq. 59). That is,

$$\mathrm{MAG}|H(f_i)| = \frac{\mathrm{MAG}|G_{xy}(f_i)|}{G_x(f_i)} \qquad (62)$$

Therefore, either Eq. (59) or Eq. (62) can be used to calculate the magnitude of the frequency response function, and as was previously noted, the phase angle of this function is identical to the cross spectrum phase angle. The magnitude of H(f) is dimensionless. No scale factors are necessary for H(f), but scale factors are necessary for the original Fourier coefficients.

Since both the PSD function and the cross spectrum function are estimates in the strictest sense, the frequency response function is also an estimate as defined and implemented here, and all of the previous frequency function constraints pertain to this function when a random noise signal is employed as the input forcing function to a linear system whose frequency response is to be analyzed. Figure 37 describes the frequency response functions, utilizing the PSD and cross spectrum functions.

The magnitude of the frequency response signal is sometimes called the transmissibility function $T_{yx}(f)$. That is,

$$T_{yx}(f_i) = \left(\frac{G_y(f_i)}{G_x(f_i)}\right)^{1/2}$$

$$= \frac{|G_{xy}(f_i)|}{G_x(f_i)} \qquad (63)$$

The frequency response function is a very powerful function which provides tremendous insight into the dynamics of mechanical systems. This function can be utilized in the random excitation control algorithm[18].

D. Servo Control of Tests[6]

1. Random Control. The essential feature of the automatic servo control system is feedback. The feedback is that property of the system which permits the output quantity to be compared with the desired reference level so that, upon the existence of a difference, an error signal arises which acts to bring the two into correspondence within the tolerance limit. In a digital random control system, a reference power spectral density in g^2/Hz is given as a function of frequency. In the case of the acoustic tests, the reference sound pressure level in dB may be given as a function of frequency. Other than the engineering units and data presentation format, random and acoustic control share the same servo concept, and the following control algorithm is equally applicable to both types of environments.

All of the system parameters that are necessary for control of a random/acoustic vibration system are not known a priori and therefore must be estimated. The driving and response signals are the only directly observable phenomena from which all control information must be deduced. The basic model for the system, as shown in Fig. 38, assumes an approximately linear transfer function for the shaker $H(\omega)$, input driving function, x(t) additive system noise n(t), and a resultant output y(t). Then,

[6] Subsections D and E were written by B. K. Kim.

$$y(t) = h(t) * x(t) + n(t) \qquad (64)$$

where h(t) is the impulse response function and * indicates convolution product. If we assume x(t) and n(t) are uncorrelated, then

$$S_y(\omega) = |H(\omega)|^2 \cdot S_x(\omega) + S_n(\omega) + E(\omega) \qquad (65)$$

where

$S_x(\omega)$ is the drive auto power spectrum

$S_y(\omega)$ is the response auto power spectrum

$S_n(\omega)$ is the system noise auto power spectrum

$E(\omega)$ is the error spectrum in estimation

Note the conventional PSD is the auto power spectrum (APS) normalized to 1 Hz as previously defined in this paper. The control problem is then to manipulate the drive x(t) or its APS $S_x(\omega)$ so that the desired output spectrum $S_y(\omega)$ is obtained.

There are several proven control algorithms which differ slightly in strategy. The following explains the two strategies.

a. Auto power spectrum control method. This method attempts to control $S_x(\omega)$ such that $S_y(\omega)$ approaches the reference spectrum $R(\omega)$ as shown in Fig. 38. This was initially proposed by Heizman (Ref. 1) and later implemented by Sotomayor(19). After computing the response signal level via APS routines, the computer compares the averaged APS to the reference spectrum and modifies the drive signal according to

$$C_i(\omega) = C_{i-1}(\omega)\left[\frac{R(\omega)}{S_{i-1}(\omega)}\right]^{1/2} \qquad C_o(\omega) = \sqrt{R(\omega)} \qquad (66)$$

where R is the reference spectrum

S_{i-1} is the averaged APS after frame i-1

C_i is the drive signal Fourier amplitude spectrum for frame i

and

$\sqrt{S_{i-1}}/C_{i-1}$ represents the latest estimate of the system transfer function.

Note the above equation involves entirely real numbers only, and the random phase is introduced just prior to the inverse Fourier transform stage. This method of control has shown in practice that it tends to require longer convergence time to a given reference spectrum, and also each individual control spectrum shows greater statistical scatter because of the random nature of the phase.

b. Complex transfer function control method(18). This method has the distinct advantage of being able to yield an estimate of the transfer function $\hat{H}(\omega)$, which is computed from the exponentially averaged cross spectrum and power spectrum as shown in Fig. 39.

$$\hat{H}(\omega) = \frac{\Sigma X^{*}(\omega) \cdot \;(\omega)}{\Sigma|X(\omega)|^2} \qquad (67)$$

Then the drive signal Fourier coefficients at frame i is

$$C_i(\omega) = \frac{\sqrt{R(\omega)}}{\hat{H}_{i.}\;(\omega)}, \quad C_o(\omega) = \sqrt{R(\omega)} \qquad (68)$$

Note in this equation that $\hat{H}$ and C are complex numbers, thus preserving the exact phase relationship. The overall performance of this method shows faster convergence and a higher statistical confidence level than the APS method. This is due to the fact that the binary psuedorandom-noise-generated phase angles are reflected in the transfer function computation process.

The digital random control presents a new concept in interpreting the control data. This stems from a unique digital processing method of time series data which is frame-oriented and a discrete Fourier transform technique which has no parallel in an analog control system. Owing to the statistical nature of a random signal, the confidence level of a PSD estimate is a direct function of the number of frames averaged and the width of a scatter interval. Assuming the random noise is Gaussian, the averaged PSD tends to a chi-square distributed function. Thus it requires a judicious choice of number of frame averages to recognize the true wideband PSD level. This will depend on the minimum confidence level desired, the maximum tolerance band acceptable, and the total duration of the test. Methods of averaging can also vary as the linear or exponential weighing may be applied.

2. Sine-Wave Control. This is the second of the main categories of control of vibration tests using a digital computer. The digital system uses a combination of hardware and software algorithms to go through all the steps of a full test. The overall block diagram has already been described in Fig. 2, and several of the functional requirements of the system have been given ample consideration above. They are:

(1) The interactive routine to input all the required test parameters and frequency profiles.

(2) The generation of the sine wave itself.

(3) The signal conditioning.

(4) The computation of the frequency and amplitude servo.

The aspect of the system in this section is concerned with the servo control concept which forms the heart of the control system for sine testing.

This control is achieved by an algorithm which basically sends out a signal at a specified frequency with a specified amplitude. This signal goes through all the conditioning steps and reaches the exciter in a smooth analog form. The analog response coming back is conditioned and digitized and its amplitude characteristics are computed. The servo algorithm then compares these amplitude characteristics at the prescribed frequency with the preset reference characteristics as determined by the input parameters. The program calculates the difference between the two levels, thus giving an error to be algebraically added to the previous amplitude of the signal sent out to the exciter, after being multiplied by a convergence factor.

The foregoing is a brief description of the amplitude servo in a sine wave test. The other aspect of the servo, the frequency servo, has already been described under the sine-wave generation section of this paper.

Before proceeding with a detailed description of the amplitude servo algorithm, it may be appropriate to consider a brief description of the different types of spectra used here. They are all amplitude-vs-frequency spectra.

a. Reference spectrum. This is the frequency profile in g's (rms, peak or peak-to-peak, depending upon the algorithm) which the test operator enters in the test setup parameters. It is the spectrum to which the servo attempts to control the response from the external load at all frequencies within the control bandwidth; i.e., it is the desired spectrum at the control point.

b. Control spectrum. This is the frequency response of the external load to the driving signal. It is the spectrum which the servo attempts to control to the reference spectrum by modifying appropriately the driving signal.

c. Monitor spectrum. This is the frequency response of monitoring acceleration signals, as selected in the test setup parameters.

d. Error spectrum. This is the difference between the control and reference spectra. It indicates by how much control is off the required specification level. It is used to correct the driving signal to the external load in an effort to bring the control spectrum as close as possible to the reference spectrum dynamically in real time. Theoretically, 100% full control would be obtained by driving the external load such that a zero error spectrum is obtained. But this results in no subsequent correction and simulates an instantaneous open-loop condition. The driving signal thus starts deviating from the value required for proper control, making the error spectrum nonzero again, and the feedback loop is closed again. So the error spectrum, as any error signal in a feedback system, is only a virtual zero: very close to zero but never being exactly equal to it.

e. Drive spectrum. This is the frequency spectrum of the signal actually sent to the external exciter. It is equal to the previous drive spectrum corrected by the error spectrum after adjusting it and properly conditioning it.

f. Description of the amplitude servo algorithm. Figure 40 shows a flowchart of the steps involved in the amplitude servo algorithm. The servo first starts at the initial frequency and outputs a sinewave cycle with an amplitude lower than the preset reference level at that frequency. It then looks at the control signal coming back, computes the error, and multiplies it by the reference spectral value and further by a convergence factor. This is then subtracted (algebraically) from the original amplitude, and the new drive signal is sent out to the external load at the updated frequency. The servo then looks again at the incoming response, corrects for the error, and outputs a further cycle. This continues until the frequency reaches the high limit of the control bandwidth.

Mathematicallv, the function of the amplitude servo in a digitally controlled sine sweep testing system can be summarized in the following equation.

$$\text{AMPL} = \text{AMPL} - \text{REF} \cdot (\text{CONT} - \text{REF}) \cdot \alpha \tag{69}$$

where

AMPL = amplitude of driving signal

REF = amplitude of reference level

CONT = amplitude of control level

= convergence factor

A word may be added about the convergence factor. The amplitude of the control spectrum is required to converge fast towards that of the reference spectrum at all frequencies. However, no overshoot (or undershoot if control was previously greater than reference) should occur. So the case is here similar to a critically damped oscillator. It is therefore at once apparent that the convergence factor depends upon the whole servo system and test article dynamics. In fact, one of the parameters in the digital sine control is the compression speed of the servo. This might be low, normal, or high, causing some small occasional overshoots (or undershoots). The value of these convergence factors is not the same (for a certain nominal compression speed) at all frequencies, i.e., α is a function of frequency. Consequently, look-up tables are generally provided within the algorithm to supply those values. It would otherwise take too much time to generate them in real-time. The dependence of α on frequency will usually range anywhere from a linear function to a logarithmic function. Figure 41 shows a typical servo speed function in dB/sec vs log frequency for the cases of a 1 to 10 gain step (marked +) and a 10 to 1 gain step (marked -). Two different compression speeds are marked High (H) and Low (L). It is seen that a faster convergence is achieved in stepdown rather than stepup jumps, and at higher rather than lower compression speed.

g. Control to a step change in reference spectrum. Since the convergence speed of the servo is finite, it is clear that programming an infinite slope step will never be controlled since this requires a very high servo speed, resulting in a high overshoot (or undershoot) which will probably exceed the abort limits preset in the test setup parameters. Steps should therefore have a finite frequency interval, so that a frequency profile switches from a level M_1 at frequency f to a level M_2 at a frequency f + Δf, where Δf is nonzero. Figure 42 shows such a change.

The Δf is a function of the magnitude of the step, the frequency at which this step occurs, the sweep rate, and the servo speed. The higher the magnitude, frequency, or sweep rate, the larger Δf must be to enable the servo to control the jump. Conversely, the higher the servo speed, the lower the necessary Δf required for satisfactory control. To calculate Δf, we proceed as follows:

Let:

N = sweep rate in octaves/min,

M = step change in voltage dB,

S = servo speed in dB/sec,

f = step change frequency in Hz

The number of octaves covered in going from frequency f to frequency f + Δf is

$$n\left[\text{oct}\right] = \frac{\ln\left(\frac{f+\Delta f}{f}\right)}{\ln 2}$$

$$= N\left[\frac{\text{oct}}{\text{min}}\right] \cdot \frac{1}{60}\left[\frac{\text{min}}{\text{sec}}\right] \cdot \frac{1}{S}\left[\frac{\text{sec}}{\text{dB}}\right]$$

$$\cdot\, M\left[\text{dB}\right] \qquad (70)$$

Solving for Δf, this yields

$$\Delta f = f\left[\exp\left(\frac{N \cdot M \cdot \ln 2}{60 \cdot S}\right) - 1\right] \qquad (71)$$

Now if g_1 and g_2 are respectively the high and low amplitudes of the two levels, then

$$M\left[\text{dB}\right] = \frac{20}{2.303}\ln\frac{g_1}{g_2}\,. \qquad (72)$$

Substituting back in Eq. (71) yields

$$\Delta f = f\left[\left(\frac{g_1}{g_2}\right)^{\gamma} - 1\right] \qquad (73)$$

where γ is given by

$$\gamma = \frac{20 \cdot N \cdot \ln 2}{2.303 \cdot 60 \cdot S} \cong 0.1\,\frac{N}{S} \qquad (74)$$

The dependence of S on frequency is typically that shown in Fig. 41. So if N, M, and f are known, Δf can be calculated. This should be used in an optimum system to calculate the convergence factor that will achieve the fastest realizable convergence without overshooting or undershooting. To this end the computer can generate a look-up table for γ once N is known from the input parameters. The reference spectrum may then be scanned and all the steps defined in terms of magnitude and frequency. A Δf may thus be calculated for each step and then used to determine the convergence factor by linear interpolation. This in turn determines what amplitude the driving signal should have.

3. Transient Control. The term "transient control test" is defined as a vibration test that induces an oscillatory pulse of short duration that satisfies physical continuity to the specimen. This is also known as "transient waveform control" (TWC). Typical durations of these tests are in the order of 1 sec or less, thus implying open-loop control. The classical shock pulses such as sawtooth, halfsine, rectangular, and triangular waveform as well as more general and arbitrary transient time histories are synthesized at a shaker control point.

Since the introduction of digital transient control techniques for electrodynamic shakers by Favour, LeBrun and Young in 1969(20), numerous laboratories have conducted transient tests using similar digital control systems. This opened a truly new capability for the digital system where there is no close parallel in the analog counterpart. Reproduction of transient waveforms on electrodynamic or hydraulic shakers is straightforeward and accurate using digital computers, provided that the test system under control is approximately linear. Recent studies of Hunter(21) and Barthmaier(13) explore the digital capability in highly nonlinear situations. The control methodology may be divided into the two following groups:

a. Specific transient time history. Given a specific transient time history, a digital system can equalize and synthesize the desired waveform, by compensating for the test syrtem transfer function(22). Software requirements are shown in a flow diagram in Fig. 43.

The main theory behind the TWC process is that of computing an instantaneous transfer function experimentally, assuming linearity of the system. It is most crucial to obtain a meaningful transfer function within the dynamic range of the digital equipment. In order to guarantee the full frequency response of wide bandwidth, a delta function can be chosen for the initial calibration pulse $f_i(t)$. In reality, an ideal delta function is simulated by a finite-amplitude square-wave pulse with extremely short duration.

The direct Fourier transform of $f_i(t)$ yields $F_i(\omega)$:

$$F_i(\omega) = \int_{-\infty}^{\infty} f_i(t)\, e^{-j\omega t}\, dt$$

$$= A\int_0^{\tau} e^{-j\omega t}\, dt$$

$$= \frac{A}{\omega}\,[\sin \omega\tau + j(\cos \omega\tau - 1)] \qquad (75)$$

and

$$\left|F_i(\omega)\right| = \frac{A}{\omega}\sqrt{2\,(1 - \cos \omega\tau)} \qquad (76)$$

The transform pair, functions $f_i(t)$ and $|F_i(\omega)|$, is shown in Fig. 44. If τ is selected as one sample period of the D/A converter, then $\pi/2\tau$ is the total band vidth of $F_i(\omega)$. The choice of a delta function as the calibration pulse is not unique, although it is found to be most convenient for digital controllability.

The calibration pulse $f_i(t)$ is transmitted to the shaker through the D/A converter, and the test specimen response signal $f_0(t)$ is detected by a transducer and stored in the computer memory through the A/D converter. A real-time Fourier transform is performed on $f_i(t)$ and $f_0(t)$:

$$F_i(\omega) = \mathscr{F}[f_i(t)]$$
$$F_0(\omega) = \mathscr{F}[f_0(t)] \qquad (77)$$

where $\mathscr{F}$ is the Fourier transform operator.

Then the transfer function is obtained through complex division,

$$H(\omega) = \frac{F_0(\omega)}{F_i(\omega)} \qquad (78)$$

Equation (78) appears to be straightforward algebra, but it requires particular care in practice. First, one must send out $f_i(t)$ such that the physical system will produce a healthy frequency response in the bandwidth of the desired transient waveform. This requires the understanding of the physical limitations of the shaker-amplifier system. It is clear that one must analyze the desired test pulse waveform x(t) beforehand to obtain its frequency bandwidth, peak acceleration, velocity, and displacement levels such that all the requirements fit within the shaker limitations. Electrodynamic shakers behave like high-pass filters up to 3000 Hz; the hydraulic shakers behave like a low-pass filter. Second, the complex division of Eq (78) must be done within the dynamic range of the digital machinery.

Once $H(\omega)$ is accurately determined, the required input waveform y(t) is synthesized in order to achieve the desired waveform x(t) at a specified location on the test specimen. The Fourier transform of the required transient wave y(t) is

$$Y(\omega) = \frac{X(\omega)}{H(\omega)} \qquad (79)$$

where $X(\omega) = \mathscr{F}[x(t)]$.

Then

$$y(t) = \mathscr{F}^{-1}[Y(\omega)] \qquad (80)$$

All the frequency functions are complex functions; thus all the Fourier transform pairs constitute one-to-one invertible mapping. This eliminates the arbitrary phase assumptions which are associated with test specifications given as frequency spectra.

Computational roundoff errors, truncation errors, and physical nonlinearity errors will occur in practice, and it is important to obtain quantitative information on the relative magnitude of the error; that is, the required function y(t) will not exactly yield the desired response x(t)

Let x(t) be the specimen response signal as y(t) is transmitted to the shaker. The error signal e(t) can be defined at each discrete point i in time as

$$e_i = x_i - \bar{x}_i, \quad i = 1, 2, \cdots, N \qquad (81)$$

N = digital frame size

The error can be given as statistical quantities, i.e., a mean and a variance. The mean of the specified waveform is

$$m_x = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (82)$$

The mean of the resulting error is

$$m_e = \frac{1}{N}\sum_{i=1}^{N} e_i \qquad (83)$$

The associated variances are

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - m_x)^2 \qquad (84)$$

and

$$\sigma_e^2 = \frac{1}{N-1}\sum_{i=1}^{N}(e_i - m_e)^2 \qquad (85)$$

Thus a measure of the error in time domain can be defined as

$$\text{percent variance error} = 100 \times \frac{\sigma_e^2}{\sigma_x^2} \qquad (86)$$

For typical transient pulses generated on electrodynamic shakers, the percent variance errors are in the neighborhood of 10%, and this is considered acceptable. TWC allows for improved control tolerances, which means that acceptable errors and tolerances can be realistically specified for this type of shock testing.

b. Shock spectrum synthesization. This requires an additional computation to determine the exact transient time history that satisfies the shock spectrum. Since a shock spectrum cannot uniquely define a corresponding time history, there are several proven methods of achieving the goal. One method(23) is to use a series of symmetric half-sine wavelets to synthesize the given shock spectrum. In due process, it allows the user to specify each wavelet's duration and time lag, thus giving control over general waveform shape in the time domain. This feature is desirable in some instances where the test specification requires time domain constraints in addition to the shock spectrum. For example, total duration and the envelope shape may be specified as well as the number of level crossings for fatigue consideration. The half-sine wavelet method also meets the shaker compatibility conditions such as the initial and final displacements and acceleration to be zero.

Each of the wavelets is an odd number of half cycles which has been weighted by a half sine wave. Each of the wavelets may also have a delay with relation to the start of the data buffer. Two individual wavelets are shown in Fig. 45 and the summation of a group of these wavelets to form a composite waveform is shown in Fig. 46. This composite waveform was synthesized to match a specific shock spectrum.

Using Yang and Saffell's notation, the composite waveform is described mathematically by the following equation:

$$W(t) = \sum_{m=1}^{M} A_m f_m$$

$$\begin{aligned} f_m &= 0 \qquad 0 \le t \le t_{dm} \\ &= \sin 2\pi\, b_m(t - t_{dm})\cdot\sin 2\pi\, N_m b_m(t - t_{dm}) \\ &\qquad\qquad t_{dm} \le t \le t_{dm} + T_r \\ &= 0 \qquad t \ge t_{dm} + T_m \end{aligned} \qquad (87)$$

where the frequency of each wavelet, $A_m f_m$, is $N_m b_m$; the frequency of its half sine envelope is b_m; and N_m is the number of half cycles. Then, T_m, the duration of the wavelet, is $T_m = 1/2b_m$ and t_{dm} is the delay of the wavelet.

A solution for the amplitudes of the wavelets, A_m, must then be found such that the shock spectrum of the composite waveform, W(t), is arbitrarily close to the specified shock spectrum. An approximation of this wavelet amplitude, A_m, is formed by dividing the desired shock spectrum value at the wavelet frequency, $N_m b_m$, by the number of half cycles N_m. This approximation is used as a starting point for an iteration procedure which calculates the amplitudes, A_m, to the desired accuracy. Accuracies of 3% or less may generally be obtained in 3 or 4 iterations. Yang and Saffell discuss some of the restrictions necessary in defining the delays, t_{dm}, the number of half cycles N_m and the spacing of the wavelet frequencies $N_m b_m$.

In one implementation, the delay and the number of half cycles are variable. This allows the operator a great deal of flexibility in defining the shape of the time history used The waveform may be made to look very much like earthquake, pyrotechnic, or other shock waveforms.

E. Test Documentation

One of the most significant advantages of a digital vibration control system over an analog system is the ability of the mainframe computer to "remember" everything that happened during the test by storing in its core memory, or on a mass storage device, the information gathered during the test run. The system is therefore able to display at the end of the test a complete documentation of the conditions under which the test was actually run. This may include important information as to the maximum deviations of the control spectrum from the reference spectrum in a random or a sine-wave test and the frequency at which it happened. It may also indicate what the test duration and actual control bandwidth was, whether any of the abort or alarm lines have sensed any abnormal conditions, and so on.

This so-called post-test documentation is especially useful when a test is aborted in an abnormal way, such as an operator manual abort, an abort due to abnormal conditions sensed on one of the abort lines, or a test specification that was exceeded during the test run. This allows the test operator to pinpoint the cause of the failure of the test and correct for it by either modifying the test specification or by checking the device connected to the activated abort line, or by taking any other appropriate measures.

Additional information pertaining to random and sine testing might be a mention of which channel was in control over what frequency range. This is especially important in multichannel sine control. Two examples showing post-test documentations of normally and abnormally completed tests are shown in Tables 5 and 6. Both are for typical sine-sweep tests. A record is maintained of the test name and/or heading, the completion status (normal or aborted and the reason for aborting the test if in fact it was aborted), and which line caused the test to abort if it was aborted. The report also shows the sweep number and frequency at which the test was aborted and the total duration of the test. Other documentation shown in these tables is the maximum and average control error (deviation of control spectrum from reference spectrum in dB) and the frequency at which the error occurred. Also shown is the control history over the whole test run, that is, which input channel was controlling, and when it was controlling, its frequency range.

Table 5. Typical post-test documentation for a completed sine test

(Test duration: 6 min; maximum absolute control error: 0.89 dB at 661 Hz; average absolute control error: 0.147 dB)

Control channel	Frequency range, Hz
	Sweep 1
1	20 - 186.3
2	186.3 - 212.1
1	212.1 - 859.1
2	859.1 - 886.5
1	886.5 - 2000
	Sweep 2
1	2000 - 882.1
2	882.1 - 855.6
1	855.6 - 210.2
2	210.2 - 184.8
1	184.8 - 20
	Sweep 3
1	20 - 186.5
2	186.5 - 212
1	212 - 859.9
2	859.9 - 886.5
1	886.5 - 2000

Table 6. Typical post-test documentation for an aborted sine test

(Completion status: aborted during sweep 1 at 247.3 Hz, abort line 5; test duration: 6 min, 12 sec; maximum absolute control error: 0.71 dB at 53.66 Hz; average absolute control error: 0.067 dB)

Control channel	Frequency range, Hz
	Sweep 1
1	20 – 186.1
2	186.1 – 211.8
1	211.8 – 247.3

V. CRITERIA FOR SELECTION, INSTALLATION, AND MAINTENANCE OF COMPUTER-CONTROLLED TEST SYSTEMS[7]

The advent of digitally controlled systems for environmental testing and analysis will substantially affect the scope, operation, and personnel structure of environmental test laboratories employing such devices. The degree of impact will depend upon the types of applications to be supported. Although only a few of these systems exist at present, dozens will be purchased in the future for test operations and control, analyses, and other related applications. The advantages of the digital test system over the older analog systems have been described in the literature(1) and are listed in a prior section of this paper and will not be repeated here except to say that the digital systems will replace the analog systems because the digital system functions are computer-controlled and are adaptable to many applications. The analytical capability of these systems can often justify their procurement almost independent of the control function.

The purpose of this section of the paper is to suggest applications, list hardware, software, and personnel requirements, and discuss pertinent questions that must be answered and facts that should be known prior to designing or specifying and procuring computer-controlled environmental test systems.

A. System Characteristics and Applications

1. Induced Random Vibration Test Systems[8] The first computer-controlled test system was designed for random environment simulation.(25) Computer-controlled sine-wave test systems, a later development, are discussed in the following section. Random-type computer test systems are apparently

[7] This section was presented originally in Ref. 24.

[8] The terms "induced" and "induced environment" as used in this section refer to a mechanically produced or simulated environmental effect, such as a mechanical vibration or shock. There is no implication that the cause of an induced environment necessarily originates in a mechanical process

creating the most interest and greatest demand at present. Unfortunately, this category of computer control has been the hardest to implement. The general cc ncepts for random test systems are described in this paper, and the basic algorithms to implement these concepts were originally presented in Ref. 19.

All of the original random-type digital control systems were designed around existing main subsystems that were available during the construction of these test systems. The basic control system hardware can be divided into three main subsystems: a Fourier processor, a computer, and an interface controller that may be part of or separate from the first two subsystems.

An interesting and probably worthwhile control concept modification was suggested by Sloane(18). In this method of random vibration control, cross-spectral density techniques are used to estimate the transfer function of the vibration system including the test item. Spectrum control is achieved at the point(s) of observation so that the output spectrum consists of two parts: a control spectrum representing the environment to which the structure is tested and a second spectrum representing uncorrelated self-noise generated from test specimen nonlinearities. Only a software change is required to implement this concept.

Computer-controlled random environment test systems should include the following multi-transducer channel functions:

(1) Spectrum or spatial averaging control.(26, 27)

(2) Alert or shutdown for signal loss.

(3) Alert or shutdown for spectrum out of specification.

2. Sine-Wave Test Systems. Computer-controlled sine test systems are less expensive than computer-controlled random test systems because the Fourier processor is not required for the sine system. However, another subsystem is required: the digitally controlled oscillator(8, 10, 11) or some other method of producing and controlling (in frequency and amplitude) a sine wave under computer control.

The role of the computer for sine-wave testing is nebulous. One manufacturer suggests letting the computer provide a supervisory role only(9) and has provided special-purpose hardware to generate and control the sine-wave forcing function. The author believes that the system computer should be made to provide and control as many functions as possible for whatever application is needed, and that this capability should be achieved through the use of software rather than by attaching peripheral hardware, which decreases the system reliability. Such a solution holds down hardware costs but increases software costs. Perhaps an optimum compromise to this dilemma is to let the computer determine and control the frequency and amplitude of a sine-wave function generator as implemented by Norin(28). In any case, a computer-controlled sine-wave test system should be able to provide the following multi-accelerometer channel functions:

(1) Peak control.

(2) RMS (or other type of averaging) control.

(3) Displacement control.

(4) Velocity control.

(5) Acceleration control.

(6) Multivibration-level programming.

(7) Signal loss alert or shutdown.

The system should also be capable of acting as a peak limiter for each channel of control, independent of the functions above. If any one, some, or all of the control accelerometers exceed a predetermined level, the system should shut itself down in a controlled manner and advise the operator which accelerometer(s) exceeded the predetermined level.

As with the random test system, the test specification for conducting a sine-wave test is entered by means of a pretest conversational language program.

At present, the additional cost factor for a sine system over an n-channel (n > 1) random system is 1.1. That is, if the random system costs 1 unit, 1.1 units will buy the sine system.

Transient waveform testing capability may add to the cost, in the form of additional software: transient waveform testing also requires a Fourier processor. If the required transient waveform is produced by the test system computer, however, no new hardware will be required If the vendor provides the transient waveform software "free of charge," then the cost factor of 1.1 applies for an n-channel random system with sine, shock, and transient waveform capability.

3. Large-Scale Data Analysis and Presentation. Large-scale time series analysis and presentation is an extremely worthwhile by-product of the induced environmental test system This type of application is made possible by the Fourier processor associated with the random-type control system (for a detailed description of this type of analysis and of the basic frequency function algorithms, see Ref. 29). The following time and frequency domain analyses are easy and inexpensive to implement:

(1) Time domain data analysis and presentation: auto-correlation, cross-correlation, and probability density and distribution

(2) Frequency domain data analysis and presentation: auto spectrum (power spectral density), cross spectrum, transfer function, coherence function, and shock spectrum

The computer-controlled test systems can support the large-scale analysis application merely by additional software and a high-speed output device that has hard copy capability.

4. Natural Environment Testing.[9] The computer-controlled test system has a definite two-part application in natural environment testing: for control and for data presentation. No new technology is required. The Fourier processor is not required for this function, but an interface controller is required, along with pressure, temperature, and humidity controllers. These controllers can control chamber environments and provide analog (or digital) information about the chamber environments. This information can be displayed alphanumerically on a cathode ray tube (CRT) and updated periodically. A permanent copy of the chamber environment parameters should be made periodically as well. Interrogation of the environmental status of any chamber from the test system teletype or CRT terminal would be a functional requirement.

Two principal factors govern the cost of a natural environment computer control system: the system as an add-on to a random and/or sine system, and the system as specifically designed for natural environment control and presentation. (The latter type of system, which is provided by some manufacturers, will not be discussed here.) The natural environment test system could be an add-on to a random system, a sine system, or a random-sine system.

The natural environment test system should be able to control and specify the following parameters:

(1) Chamber pressure control.

(2) Chamber, fixture (heat exchanger), and test item temperature control.

(3) Chamber humidity control.

(4) Specification of alarm limits for pressure, temperature, humidity, chamber line voltage and power interruption, test item power supply voltage, ambient room temperature, and water, air, and nitrogen pressures.

The following parameters should be available for presentation:

(1) Chamber pressure.

(2) Foreline pressure.

(3) Backing pressure.

(4) Chamber, fixture, test item, and ambient temperatures.

(5) Chamber humidity

(6) Pump status: on/off and valve positions.

(7) Firm power and test item power status.

5. General-Engineering Use. The basic computer-controlled test system makes an excellent general-engineering tool for solving those types of engineering problems associated with the environmental sciences, including mechanical and electrical engineering analysis and synthesis. This capability results from the general-purpose computer associated with these systems. All of the computers used thus far will support the BASIC user-oriented language, and some computers used can also support FORTRAN IV or a form of FORTRAN, although not conveniently without some sort of mass memory storage. Therefore, it is recommended that mass memory be considered for this application and the application to be discussed below.

An alternative to the mass storage hardware is interfacing the computer to a larger computer system. This is called multiprocessing and makes the control system computer an extremely powerful tool at very little additional cost. Computer-to-computer (also called processor-to-processor) data exchange provides two computers, each with local file space, to communicate in such a way that each computer may access data stored in the other. This allows the control system to perform such tasks as interactive fixture design and to perform system analysis. However, 12,000 words of memory will allow engineers to do most of the run-of-the-mill types of mechanical and electronic analysis and synthesis problems required in test and control.

6. Laboratory Management Use. The basic control system used as a laboratory management tool requires mass storage capability or computer-to-computer interfacing. The term "laboratory management tool" is used to denote a method of implementing the mathematics of system analysis, which includes (but is not limited to) the following management sciences:

(1) Linear, nonlinear, and dynamic programming

(2) Reliability studies.

(3) Statistical analysis and probability theory.

(4) Decision and game theory.

(5) Queue theory, including test and design scheduling.

(6) Information theory.

(7) Forecasting.

(8) Program evaluation and review technique (PERT) (critical path method) and networks.

(9) Human engineering

(10) Financial analysis

References 30 - 32 discuss most of these topics in a general way. Computer programs have been written to implement the algorithms associated with these management tools

The control system computer and input/output (I/O) devices can be used for general administrative tasks such as accounting and inventory and

[9] The terms "natural" and "natural environment" as used in this paper refer to environmental effects that occur in nature, including temperature, vacuum, humidity, sunlight, sand and dust, fungus, salt spray, or any combinations of these.

equipment control. The system can also generate test procedures and modify or update these procedures, as required, since the computer manufacturer includes in his software a means of updating and modifying source programs on a line-by-line basis. This means that long lists and text can be modified more easily and quickly by the computer than by a typist.

7. Other Applications. Only the creativity and imagination of the vendor and the potential user limit the applications of computer-controlled test systems. Besides the above-listed applications, these systems are being used as transducer calibrators, earthquake simulators, road simulators, mechanical impedance analyzers, and nuclear reactor monitors.

The number and type of applications will determine the size, complexity, and cost of the computer-controlled operating system. (An operating system is defined here to mean a test system under control of an executive program such that the operator merely specifies the test function(s) desired. The system does the rest, obtaining information from the operator as required.)

Some of the applications above can be (and may be required to be) supported simultaneously. For example, the natural environment test control functions, as described above, can be carried on in parallel with an induced environment test. However, if several parallel operations are necessary, two (or more) computers may be required.

8. Criteria for Determining Applications. How does one recognize the requirements for computer-controlled systems? The answer, of course, is in knowing one's own testing requirements and knowing the advantages and capabilities of computer-controlled test systems, as well as carefully considering the following factors:

(1) The test function performed by the environmental test laboratory. If the function is to provide induced random and/or sine-wave simulation and/or shock pulse or shock spectrum environments to complex test items requiring complex fixturing, then the laboratory may be a likely candidate for a computer-controlled test system. If the test program requires testing m items using n test specifications, where m is a relatively large number, and there is truly a need to ensure test repeatability by implementing the n specifications several times during the test program, there is an additional argument for the computer-controlled test system. If, in addition, any appreciable amount of data reduction, analysis, and presentation is required for test documentation or test specimen design, development, and reliability or structural behavior studies, there is all the more reason for considering computer-controlled test systems. If the laboratory is also performing natural environment testing on the same m items, there may be a need for an operating test system to provide all the test functions.

(2) The requirements that may make the system feasible. Two conditions under which a computer-controlled test system may be justified are: when the procurement or construction of an automated test system provides the laboratory with greater test control features and when operating costs may be reduced, thereby achieving a less expensive program. The following are additional considerations in making such a choice:

(a) Increase in test volume (that is, a greater number of tests in a given period of time) for the same test facilities and space.

(b) Safer and more accurate testing with reduced setup time and fewer human errors.

(c) Repeatable testing and testing that can be exactly duplicated at other laboratories.

(d) Data analysis capability without depending on external (department, plant, outside agency) facilities.

(e) Automatic post-test performance documentation.

(f) Flexible capability of implementing new or different test philosophies and expanding the test capability merely by changing or modifying software.

(3) The existing test system control capability and the system's limitations and problems. If the existing system is adequate for the present and near-future requirements, it may not be possible to justify a computer-controlled test system at present. On the other hand, arguments for justifying the computer may lie in situations where, (a), it is necessary to subcontract such functions as fixture design and data reduction, analysis, and presentation; (b), the laboratory is unable to do all the testing that is required; (c), the tests are based on what can conveniently be done, rather than on what should be done; or (d), the addition of a new system would increase the test control capability.

(4) The possibility that a new control system must interface with any part of the existing control system(s). A computer may create new control problems if it must be interfaced to existing hardware. On the other hand, the old system can possibly be used in its existing form as a parallel redundant (but limited) test control system, completely independent from the new system.

B. System Hardware and Software

1. Add-on Hardware and Software. Table 7 lists the hierarchy of hardware and software for the types of applications that can be supported by computer-controlled environmental test systems on an add-on basis.

Table 7. Computer test systems hierarchy of hardware and software

Application	Requirements
I. Basic system, n channels of test control	A. Computer system 1. General-purpose computer 2. High-speed paper tape reader 3. High-speed paper tape punch 4. Analog multiplexer and A/D converter 5. Priority interrupt lines 6. Direct multiplex control/direct memory access channels 7. Group I software package (equipment manufacturer's software library) 8. 8000 core memory B. Conditioning and conversion equipment 1. Input/output amplifiers/attenuators 2. D/A converter 3. Input anti-aliasing filters 4. Output low-pass filter C. Input/output equipment 1. CRT/keyboard terminal with hard copy capability 2. Group I cable and connector package
II. Random system, n channels of test control (for induced vibration and acoustic testing)	A. Group I hardware and software B. Fourier processor (dual channel) C. Interface controller, version 1 D. Additional 4000 memory (12,000 total) E. Group II software package F Group II cable and connector package
III Large-scale data analysis and presentation	A. Group II hardware and software B. Group III software package
IV. Sine (n channels) shock and transient waveform analysis and control	A. Group I hardware and software B. Additional conditioning and conversion equipment 1. Computer-controlled function generator 2. Peak control hardware 3. Average control hardware C. Interface controller, version 2 D. Group IV software package E. Group IV cable and connector package
V. Natural environment test system, n channels (less chamber controllers)	A. Basic system B. Multiplexer expansion for computer output control C. Real-time clock added to computer D. Group V software package E Interface controller, version 3 F. Group V cable and connector package
VI. General engineering tool	A. Basic system B Additional 4000 memory (16,000 total) C. User-generated software
VII. Laboratory management tool	A. Group VI hardware B Mass storage capability or link-up capability to large-scale computer C. Additional equipment manufacturer's software D. User-generated software E Optional high-speed input device

A basic system is defined in terms of three elements: the computer, conditioning and conversion equipment, and input/output devices. These elements, consisting of both hardware and software, are required independent of the application and, whatever applications are required, additional hardware and software must be added to the basic system.

The next system in Table 7 is defined for induced random-excitation environmental testing. This application was chosen second to the basic system, rather than a sine-wave test system, because of present interest and demand. The large-scale data analysis test system is essentially a by-product of the random test system and requires only some additional core memory and software. The sine-wave test system also requires additional hardware and different software but does not require the Fourier processor used in the random test system However, it may be possible to utilize the Fourier processor for sine-wave testing at some future time since the hardwired Fourier processors contain sine (and cosine) tables within "read only memory" in order to implement the fast Fourier transform algorithm. A different version of the interface controller is required for thr applications of random and sine test systems as well as a natural environment test system. These different versions are really insignificant in terms of technology and complexity, and could easily be incorporated into a single controller at very little if any additional cost.

The main additional hardware for the natural environment test system is an expansion of the multiplexer capability in order to control n number of chambers and m functions for each chamber. In addition, a real-time clock is necessary in this application in order to conveniently present the chamber parameters (environments) at regular intervals of time.

The test systems used as general engineering and management tools require only additional computer memory storage capability and specialized software, mostly user-generated.

2. Control System Software. 'A key question after the applications are chosen concerns the status of the required software; that is, are the software logic programs available to satisfy the proposed applications? The specific applications? Or will the vendor develop this software while he is putting the system together? Or should the user's programmer develop it?

The software logic programs have been developed for "standard" random sine-wave control, and transient waveform control and analysis. Shock spectrum testing philosophy is as confused in the digital world as it is in the analog world.

To the best of the authors' knowledge, there has been no software development for natural environment testing by the vendors in the business of supplying these new computer-controlled environmental test systems, but some software development has been done by vendors supplying computer control systems specifically for chamber control. As special applications appear, special programs will be developed. If every vendor used the same equipment, a program bank could be established, but this is not likely to happen. The closest approach to such cooperation thus far on these systems is the use of the Digital Equipment Corp. PDP-11 computer, although this is not universal.

One factor is common to all the present control systems: the logic programs are generally written in assembly language and, when these systems are procured, the buyer receives system tapes to implement the control functions. These system tapes were generated by the vendor using the procedure in Fig. 47. If the user's organization structure includes a programmer, as the author recommends, then the control system specification must include the source tapes used in generating the system tapes, whether the tapes are proprietary or not.

All of the post-procurement programs should be written in the assembly language supported by the system assembler. (Higher-level computer languages such as FORTRAN are probably too inefficient for real-time control.) The vendor should supply updated programs and listings free of charge. The computer manufacturer will send the user updated object tapes and source listing for the computer. There may be a charge for these programs.

C. Constraints and Tradeoffs

1. General Specifications. The system is specified when the applications are defined. However, there is more than one vendor available to build an operating system, and each vendor has a seemingly endless list of options. This section discusses some important parameters of the system and some of the input/output and memory options, along with other constraints.

The five most important requirements of the total system are, in order of importance:

(1) The system must be flexible enough to meet current and anticipated needs.

(2) The system must be safe to operate and work around.

(3) The system must be human-engineered for ease of operation, maintenance, and personnel training.

(4) The system must be adaptive and easily and economically expandable, both from the standpoint of hardware and of software.

(5) The system must have a practical capital cost and practical cost of operation.

2. Input-Output Options The I/O options probably deserve the most concern and contribute significantly to the overall cost. Table 8 lists some of the more popular I/O devices. A choice may have to be made between a teleprinter such as a Teletype Corp. ASR 33 or 35 and a CRT terminal such as a Tektronix 4010-1 with the 4610 hard copy unit (The designation "ASR" means "automatic send-receive.") The 33 machine, which comes "standard" with most of the computer-controlled test systems, is made mostly of plastic and is inexpensive. It is a good machine for periodic use, but it is not a continuous-duty machine. A user who needs a continuous-duty machine should specify the ASR 35, which is made of metal and is more than twice as expensive as the ASR 33. On the other hand, the 33 is more convenient to use than the 35 for correcting and duplicating short programs. The author recommends an ASR 33 if the user is specifying a high-speed reader and punch and if he plans to obtain some other high-speed output device. A user who is teletype-limited should specify the 35 over the 33

These teletypes (TTYs) will input data to the system from either the keyboard or prepunched paper tape (they have slow-speed feeler wire-type

Table 8. Input/output devices suitable for computer-controlled test systems

Device	Cost factor	Functions		Significant limitations	
		Input	Output	Input	Output
TTY M33	1.0	Keyboard, paper tape reader	Serial printer paper tape punch	Slow Not continuous duty	
TTY M35	2.4	Keyboard, paper tape reader	Serial printer paper tape punch	Slow	
CRT terminal with hard copy capability	4.0 and up	Keyboard	Alphanumeric CRT display with hard copy capability	No tape reader	No tape punch
High-speed paper tape reader	2.0	Paper tape reader	–	–	–
High-speed paper tape punch	1.8–2.5	–	Paper tape punch	–	Cost function is proportional to speed
Combination high-speed reader and punch	2.0	Paper tape reader	Paper tape punch	200 characters/s	50 characters/s (not really high speed)
Plotter	0.7–5.0	–	Alphanumeric graphic presentation	–	Interfacing and D/A conversion if plotter is analog type
High-speed printer	3.0–10.0	–	Serial or parallel (line) printer	–	Requires cache storage for full speed advantage

readers) and will print out data from the system onto paper tape and punch the data onto tape. It is also possible to punch a tape from the keyboard off-line: that is, when the TTY is not electrically connected to the system.

The TTYs are slow (about 10 characters/sec) and, except for very short source programs, are virtually useless for a paper tape input device.

A low-cost CRT terminal costs slightly more than four times as much as an ASR 33 TTY. This graphic terminal with the hard copy option will do everything the TTYs do and more, except that it will not punch or read tape. The hard copy option will allow permanent copies of the CRT display. The first copy takes about 18 sec. Test specifications can be placed into the operating system via the CRT terminal keyboard. An optional TTY port interface is required with the Tektronix unit mentioned above. The hard copy option cannot be added to the standard terminal after the fact. If a user will need the hard copy capability at a later date, he must specify that option when specifying the initial CRT terminal.

The graphic display capability of the CRT terminal is the plus factor over the TTY. Software logic programs supported by the control system assembler are available from the CRT terminal manufacturer for a small fee. These programs conveniently allow the systems operator to utilize the full capability of the graphic display.

If the user's organization will include a programmer, as recommended, then a high-speed paper tape punch will be required for tape program generation. (A high-speed paper tape reader comes with the control system.) High-speed punching is from 50 to 110 characters/sec. (High-speed paper tape reading is from 150 to 300 characters/sec.)

It is very desirable to have some type of high-speed printer for obtaining source program listings when developing new software logic programs. However, the CRT terminal could be used for the source listings, producing one page every 18 sec, 72 characters/line, and 35 lines/page. A line printer operating at 300 lines/min, 120 characters/line, and 50 lines/page is faster. The decision must be based on the amount of programming required and on the applications. Another consideration, however, is the generation and cost of multiple copies, in contrast to single copies. Impact types of printers can make as many as five copies simultaneously, and TTYs can make two copies simultaneously. The CRT devices can make only one copy of the display at a time, but, of course, it is possible to copy the display as many times as necessary.

Plotters can also be considered as a means of graphically displaying the results of a test or the output from the system. There are many devices, both analog and digital, that could be used. The plotter must be controlled by the computer. It should also be able to plot and label the output from an analysis program within, say, 20 sec or the system will usually be waiting for the plotter. As it is, the I/O devices are the slowest links in the control system.

Input amplifiers and anti-aliasing filters, although not truly considered I/O devices in the computer field sense, are important analog input devices, as discussed previously. Transducer conditioning amplifiers should be of the differential-input type in order to cancel out common-mode noise riding on the shielded cables from the transducers. There should be at least 90 dB of common-mode rejection at the nominal gain settings of these amplifiers. They must not be slew-rate limited for shock testing.

The input anti-aliasing filters set the control bandwidth commensurate with the sampling theorem. The low-pass filters should be capable of at least 36 dB/octave rolloff from the cutoff frequency (6-pole filters). Their characteristics should be of the Butterworth type (maximally flat response) for vibration control, but they must have linear phase characteristics for shock test work: that is, both shock spectrum and transient waveform types of testing.

3. Control Loop Cycle Speed Constraints. How fast must the internal processes of the control system be performed in order to implement the application concept? This is difficult to answer, but the question can be restated: "Are the internal processes fast enough to meet the application?" The speed requirement is of interest only when one is using the system for real-time control or real-time analysis and, even then, is critical only for random-type testing Some of the latest special-purpose Fourier processors are so fast that the memory cycle time of the computer is becoming the limiting speed factor. Speed is important because a minimum number of algorithms must be executed by the system in order to implement any control application. It is desirable to have some time left over so that additional algorithms or additional coding within an algorithm can be added to the program

a. Sine-wave testing. If special hardware is used to generate a swept sine wave, and special hardware is used to implement the concepts of peak or average control of several control channels, then no particular speed concern is required. However, if some of these functions are performed by the computer, then it is the vendor's responsibility to assure the customer that the processes can in fact be carried out fast enough so that his system will meet the functional requirements and specifications. (Another constraint on the vendor is that his algorithms are valid processes, mathematically speaking, and generate physically valid results under the concepts of environmental testing.)

b. Random testing. The present-day computer-controlled random test systems are marginally fast enough to implement the real-time control function. In fact, it has been necessary in some systems to repeat, sequentially, several output data frames (time domain output forcing functions) in order to provide enough time to perform all the algorithms necessary for control. This repetition, if done properly, is legitimate from the mathematical standpoint and probably does not degrade the overall induced random test philosophy, although it results in a significantly different audible noise from the shaker.

The random test loop cycle time is affected by analysis parameters described in a previous section of this paper. The random system used as an analyzer gives the customer a large combination of effective filter bandwidths and analysis bandwidths, but, used as a controller, the system allows only a limited number of choices because of the time required to execute the logic programs during the control loop cycle.

The number of multiplex control channels is limited by the computer and multiplexer and by the maximum A/D converter sample rate. Eight-channel control with analysis bandwidths up to 2500 Hz is available. The number of control channels can always be increased by spending more money, and 30- or 40-channel multiplexing is feasible but expensive. Another problem, however, has only a philosophical answer. The problem is defined as follows. Suppose 30-channel control is required for an induced environmental test. Suppose also that we sample every 200 μsec and generate 100 spectral lines (200 frame size) for an analysis bandwidth of 2500 Hz and an effective filter bandwidth of 25 Hz. How long do we sample each response transducer before moving to the next? If our control loop cycle time is 80 msec and we sample each transducer for 1 input frame (40 msec), it would take 2.4 sec to cycle through all the transducers. Or should we sample the first transducer the first 200 μsec, the second transducer the second 200 μsec, etc.? In short, what kind of spectral average control is proper for the test?

More control loop cycle time is available for reverberation acoustic testing because of the response time of the reverberation chamber, particularly chambers larger than 140 m^3 (~5000 ft^3). Repeating output frames is desirable in this case because the time constants of large chambers are in the order of seconds at frequencies below 500 Hz. External generation of energy above 1000 Hz is usually not required. Convergence to the desired constant bandwidth test specification must be very slow compared to random vibration tests, or instability results.

c. Natural environment testing. Operating systems as described in this paper can easily control and present environmental data from at least 24 chambers. This is an easy task for such a system The computer would be busy for only a few milliseconds each minute or so, unless it was interrogated by the systems operator or unless a TTY rather than a CRT terminal was used for data presentation.

4. Storage Options. The amount of core memory specified will depend on the applications for the control system. In any case, 16,000-word storage should be the maximum upper limit. If more memory is required, slower-speed mass storage should be considered (mass memory storage access time is slower than core memory access time). Core storage is required for real-time control of sine-shock, random, and natural environmental testing. Mass storage can be a convenient or necessary storage supplement for the general engineering and management tool applications.

If applications justify a need for mass storage capability, the user should specify at least 20 times more mass storage than would be specified for core storage. Mass storage is available in the form of drums, discs, or tape. Table 9 lists and defines several types of mass storage devices suitable for the applications defined in this paper. The tape cassette and cartridge mass storage devices are becoming very popular for minicomputer systems because they are less expensive than any of the other devices. For the same mass storage capability, a tape cartridge unit will cost only about one quarter as much as a fixed-head disc, based upon present prices. The trend in the minicomputer business seems to point toward tape cassette and cartridge mass storage. However, a review of Table 9 reveals the tradeoffs required for the lower-cost tape units.

D. Reliability and Maintenance

1. Control System Reliability. The general reliability function R(t) of the control system is given by

$$R(t) = \exp\left[-\int h(t)\, dt\right] \tag{88}$$

where h(t) dt is defined as the conditional probability of a failure in the interval t, t + dt, given failure-free performance up to time t. The function h(t) is called the integrant hazard function, or instantaneous failure rate, or, sometimes, the force of mortality. Unfortunately, h(t) is unknown at the present time. If it is invariant with time, then h(t) = λ (a constant) and h(t) is called simply the failure rate, in which case,

$$R(t) = \exp(-\lambda t) \tag{89}$$

It should be understood that there is no indication that the control system reliability model has an exponential form. Two years of experience with a computer-controlled random system suggests, in fact, that the failure rate is variable. But in any case, it is impossible to give mean time between failure (MTBF) or mean time to repair (MTTR) figures for these systems. It is possible to obtain reliability figures from the computer manufacturer and perhaps the Fourier processor manufacturer, in the case of a random-type control system. A simplified series bimodal (failure by open or short) reliability model could then be constructed. What really is needed is the failure density function, which is defined as the probability density of the occurrence of failure as a function of time.

2. Maintenance of the System. The purchasers of the computer-controlled systems are obligated to maintain and repair them. Some or all the responsibility can be transferred to the vendor in the form of a service contract, or the responsibility for different parts of the system can be transferred to outside service agencies. The authors recommend in-house support capability with a maintenance contract on the TTY, and vendor and manufacturer support, when required, with no service contract. The rationale for these recommendations is developed below.

Thus far, the available control systems as described in this paper are available from vendors who manufacture only part of their control systems, although this will not always be the case. At present, if a vendor buys a computer as an original equipment manufacturer (OEM), will the vendor be able to maintain and repair it as well as the computer manufacturer? Hardly. It would probably be better (less expensive) to call for the computer manufacturer's services when required. It is doubtful that the vendor, as an OEM, will ever be able to really compete with the computer manufacturer in servicing the system computer. At

Table 9. Memory storage tradeoff parameters

Storage device	Cost factor	Words storage per block	Average access time	Transfer rate, time/word
Core	1.0	4,000 words 16,000 words max[a]	1.2 μsec	1.2 μsec
Fixed head disc	4.0	64,000 words 256,000 words max[a]	17 msec	16 μsec
Tape cartridge	0.8	64,000 words/track[a] 4 tracks max 90 m (~300 ft) max	60 sec 23-m (~75-ft) tape at 19 cm/sec (7 5 in/sec)	3.55 msec

[a] Maximum as defined in this table means a practical upper limit for the applications described in the paper.

best, a service agreement between the vendor and the manufacturer of the computer could be arranged. The same reasoning applies to the Fourier processor, if one is required.

The vendor can service and maintain the smaller, less expensive peripheral components, even though he may not manufacture them. One or two exceptions might be such items as disc or drum memories, line printers, and perhaps CRT terminals.

Another problem may be the location of the vendor's service facility. How far is the facility from the user's location? Will someone from that facility be able to service the system or will the factory be required to send someone?

The minicomputers used for test control can be serviced in much the same way as the old black-and-white TV sets. First, one reads in the diagnostic programs to determine what part of the computer is malfunctioning. (In the case of the TV set, the symptoms could be seen on the TV screen.) Then one can start plugging in spare logic cards, analogous to the spare TV tubes. With luck, the problem can be found. Reliability studies show that the integrated circuits are the least likely devices to fail. But the point is, the system analyst (if there is one) has a good chance of finding and correcting a system failure, depending upon his experience and training. Several minicomputer manufacturers offer maintenance classes either free (for their customers) or at a nominal charge.

The computer is not the only subsystem that may cause problems. The computer, the Fourier processor, and perhaps the interface controller are the most complex subsystems, however, in the control system.

One precaution is important: if any type of service is to be attempted by anyone, manuals, schematics, procedures, and diagnostic programs will be required. The user should specify them and see to it that they are kept up to date (as the system analyst's responsibility)[10]. The vendor has a fundamental obligation to make this documentation available, to make sure it is accurate, and to keep it accurate as field modifications to his system are made. This does not mean, however, that he should supply this documentation free of charge.

Preventive maintenance which involves more than simply repairing hardware should be an in-house support capability, except for the TTYs. Teletype machines are mechanical monsters and require careful looking after. They need to be almost immersed in oil to operate. Parts wear out and clutches need adjusting, springs need pulling or cutting; above all, TTYs must be kept clean. Although great care has been taken not to mention actual sums of money in this paper, it is important to know that a TTY will cost from $120 to $300/year to maintain, depending upon how many machines a company is having maintained on a service contract. The cost will also depend upon the part of the country in which a user is located.

Table 10 lists some of the preventive maintenance functions required for an all-round preventive maintenance program for random-type control systems.

Table 10. Some preventive maintenance functions for computer-controlled random test systems

Period	Maintenance function
Daily	1. Check cooling fans 2. Run control system integrity diagnostics
3 months	1. Check and adjust master clock frequency 2. Check all power supply voltages 3. Inspect, clean and oil TTY 4. Clean all air filters 5. Clean and oil high-speed punch 6. Align optical paper tape reader 7. Clean and adjust line printer
6 months	1. Calibrate all rms and peak reading meters 2. Calibrate oscilloscope and CRT displays 3. Calibrate charge amplifiers 4. Calibrate I/O amplifiers and low-pass filters 5. Calibrate A/D converter(s) 6. Calibrate D/A converter(s)

If the user plans to do most of his own maintenance (as recommended), it is very important to consider spare parts and diagnostic programs.

a. Spare parts. What parts of the system will be available from the vendor at a moment's notice? Will it be less expensive to obtain parts from the manufacturer rather than the vendor? (The vendor may also be the manufacturer.) What about subsystem parts such as power supplies, converters, TTYs, etc.?

It probably will be worthwhile to stock some logic cards for the computer, interface, and the Fourier processor, if the system has a Fourier processor Some computer manufacturers sell maintenance and spare parts kits. Of course, this will add to the total cost of the control system. Other spare parts and supplies should include:

(1) Ink ribbons for TTY and printer.

(2) Paper for TTY and printer or hard-copy machine.

(3) Paper tape for TTY and high-speed punch.

(4) Oil and grease for paper tape punch and TTY.

[10] Personnel responsibilities in general are discussed in later sections.

(5) Blank format tapes for high-speed printer.

(6) Logic probe (very handy for troubleshooting).

(7) Spare panel lamps.

(8) Tape winder (this device is an absolute necessity) for other than fan fold tape.

b. Diagnostics. If the vendor spends x hours in the development of software logic programs to meet an application, approximately 3x hours should be spent in developing diagnostic programs to determine whether, in fact, the application is being met, and if not, why not. This is a very important point, perhaps the most important point of this paper. When the system is not working correctly, how does one know what is wrong? Only the effect is observable. The cause is hidden. A properly organized diagnostic program will allow the system to diagnose itself. The computer manufacturers have diagnostics for their computers and their computer options. They have expended much money and effort in this type of software development. It is necessary for the vendor to do likewise. If the vendor refuses to develop good diagnostic programs, the user must write his own, in which case he would require considerable knowledge of the vendor's hardware and software. The responsibility of writing diagnostics should fall on the vendor. It is for this reason that a good control system specification must include diagnostic requirements. The vendor certainly has the right to charge for this very important software development.

Table 11 lists the type of diagnostic programs required to maintain a digitally controlled random test system.

Table 11. Recommended diagnostics, computer-controlled random test system

Computer[a]	Core memory test Central processing unit (CPU) test Paper tape reader and punch integrity Real-time clock test and calibration Line printer integrity
System	Algorithm integrity (dual channel) Internal data transfer test Core memory test (Fourier processor) Frame size selection Data transfer across interface, I/O test Control functions test Interrupt handling functions test Timing[b]

[a] Usually supplied by computer manufacturer.

[b] Requires logic hardware and oscilloscope.

It is also very desirable to write some sort of timing diagnostic. It seems there is no good way of doing this without the help of some logic hardware. Logic hardware should be included within the interface controller in such a way that, with the aid of an oscilloscope and a software logic program, the system timing can be mapped and diagnosed.

E. Facility Requirements

Some facility changes may be required in order to accommodate a computer-controlled test system in a testing laboratory. The test system should not be located close to any strong electromagnetic interference source such as power amplifiers, shakers, shock machines, or cycling vacuum pumps. In addition, the equipment should be located in an air-conditioned environment where there is a minimal change in temperature throughout the testing day. It should be a clean environment; the user should inspect his own plant's computer facility and observe that environment.

These control systems will require input power from 3000 to 6500 W, which means a minimal current requirement of 25 A. The power source should be dedicated to the control system. That is, a separate transformer should be installed, and the control system should be the only equipment connected to the secondary of that transformer. Multiple ac outlets must be provided from the power source. The dedicated transformer need not be of the isolated type.

The line voltage, 117 Vac, must be constant, or computer problems will result. If the line voltage drops too low, the system computer will not operate properly, although no one may be immediately aware that it is malfunctioning.

If the control system is to be operated from a remote location, a cable raceway will be required from the test site to the control system. Only signal lines should be laid on this raceway. If power switching will be performed from the control system, the power switching lines should not be routed parallel to the signal lines.

Television monitoring equipment and some means of audio communications may be required for remote operation.

It may be necessary to reconfigure some existing equipment in order to accommodate the control system. All these costs must be considered when planning for the control system. If the control system is to be coupled to an existing computer facility, additional hardware including data sets (modems) and cabling will be required, increasing the total cost.

F. Personnel Requirements

The use of a computer in a testing laboratory will require new or retrained personnel if the manager is to avoid a drastically increasing operating budget; it is not financially feasible to have the vendor in residence at the laboratory.

Moreover, consider the differences between analog and digital test systems: the analog systems consist of hardware performing a finite number of functions, while the digital systems consist of both hardware and software and can best be described as concepts. If necessary, one can change, modify, or add to the software of a digital system—in other words, change the concept of the system. This cannot be done with the analog systems currently available. To make the best use of this adaptive capability, the manager requires personnel skilled in the applications of computer control and knowledgeable in the specific kinds of testing to be done. At least part of the testing organization will be fashioned into a small computer center.

Figure 48 illustrates the relationship between an operating system and the skills required for an induced environment test group. These are working group functions required for performing induced environmental testing and in no way reflect the structure of the entire test laboratory. (Of course, one person may perform two or three different functions.) Some of these necessary personnel functions are further discussed below.

1. Data Manager. The data manager (who is not the test laboratory manager) is in charge of and responsible for the functional structure illustrated in Fig. 48. He must have a keen understanding of both induced environmental testing and the computer sciences as applied to his operating system. His background should be mechanical as well as electrical (electronic) and must include a comprehensive understanding of the stochastic processes. He must also have a good understanding of test operations.

2. Programmer. A programmer is required to change and add software logic programs as required. The vendor cannot be expected to make minor programming changes or write and assemble programs without a handsome charge.

All the programs required by the operating system are written in assembly language or other special language, not in a symbolic user-oriented language such as FORTRAN. The computer manufacturer supplies an assembler with his machine to the vendor. In addition, a "complete" library of programs written by the computer manufacturer is also supplied, usually free of charge, to the vendor. The vendor, whose task is to put a system together, uses some of these programs and some of his own to make up the tapes required for the operating system. Figure 47 shows the procedure required to make a system tape. It is the programmer's job to make system tapes, as required.

If the programmer is to implement the laboratory's special requirements, he will surely need records of all the vendor's and computer manufacturer's source listings. However, the vendors are reluctant to supply listings with their systems because they believe their ideas should be proprietary. (The programmer could, with a good deal of trouble, decode the systems tape anyway.) The point is that when a computer-controlled test system is specified, the user should demand listings of all the source programs used in making up the operating systems tapes.

The programmer should know assembly language programming as well as those user-oriented languages that can be supported by the system computer. He should be extremely knowledgeable in the physical processes he is trying to implement: in this case, vibration, shock, and acoustic testing and analyses. He must have such a clear understanding of the operating system that he can change its concept if required.

3. System Analyst. The function of the system analyst is to analyze and solve problems in the hardware and to help the programmer debug and analyze the results of his programs. All programming modifications and additions must be approved and recorded by him. The system analyst must be kept aware of every hardware and software change. It is his responsibility to record and log all system anomalies and his decision on whether vendor or computer manufacturer help is required. He controls the required inventory on spare parts, including logic cards, lamps, TTY ribbons and paper, line printer ribbons and paper, paper tape, and whatever else is required. He is in charge of a preventive maintenance plan for the entire test system. He is responsible for keeping all software logic program documentation updated, including vendor and computer manufacturer software updates and engineering field modifications. His educational background should be in the computer sciences and he must have a good working knowledge of the processes implemented by his operating system.

4. System Operator. The operator is responsible for operation of the computer system and for seeing that the system implements the required concepts. He inputs the test specifications via the conversational language programs or prepunched tapes. He is responsible for implementing the test specifications and obtaining test results for the test engineer. He notifies the system analyst in case of trouble and works with the system analyst and the programmer. He is responsible for running the routine system integrity diagnostics executed prior to each day's testing. His educational background should include some assembly level programming and induced environmental test operations.

5. Test Engineer. The test engineer needs additional knowledge in the computer sciences and considerable knowledge in the field of time series analysis. He must be able to determine requirements, procedures, and processes to implement an environmental specification and accurately analyze the results.

6. Training. Training personnel to understand and operate a computer-controlled test system should be begun before taking delivery of the system. The training can come from several sources, including the system manufacturer (the vendor), the subsystem manufacturer, an out-of-plant educational facility or organization, or in-plant training facility. The results of the training (there should never be a final results) should provide answers to the following questions:

(1) What is the system supposed to do?

(2) Is the system doing what it is supposed to do?

(3) How is the system doing what it is supposed to do?

(4) Why is the system doing what it is doing?

(5) What can be done if the system is not doing what it is supposed to do?

(6) How can the system be made to do something it is not supposed to do?

The best way to obtain the answers to the questions above is obviously to design and build one's own control system, using available subsystems as building blocks. The next best way to obtain the answers is to become knowledgeable in the following disciplines:

(1) Environmental testing and current test philosophies.

(2) Engineering mechanics (structural dynamics), physics, and vacuum technology.

(3) Electronic engineering: analog and digital design and analysis.

(4) Computer system architecture and organization.

(5) Control system analysis and synthesis.

(6) Digital communications and information theory.

(7) Assembly language programming.

(8) Time series analysis.

(9) Probability theory and statistics.

(10) Data management.

The amount of training required to meet the organization's needs will depend upon whether existing test personnel are trained or new people are brought into the test organization. If existing test personnel are trained, they will probably be versed in the disciplines of items 1 and perhaps 2 only, requiring some training in the remaining disciplines. On the other hand, bringing new people into the test organization may require less training, provided these people have the proper educational background and/or work experience. For example, a new computer science (or information science) graduate will probably require training in the disciplines listed from items 1, 2, and 5. A recent electronics engineering graduate should have a good working knowledge of most of the above-listed disciplines except perhaps those covered in items 1, 7, and 10. If new personnel must be added, it is desirable to hire recent graduates from the appropriate sciences and then train these people as required, because the technology in this country is probably still doubling every four years and, in digital electronics and computer field, every two years. Recent graduates provide updated knowledge, which can be advantageous to any organization.

It is the test laboratory manager's responsibility to see that his people are qualified and kept properly trained to perform their work functions. Education is a never-ending requirement in any technical discipline, including environmental testing. Environmental testing is a philosophy, not a science but the means by which the philosophy is implemented is a science and requires scientific procedure.

The bulk of the training burden should be placed on the vendor, as it is in the computer business. Some training can be obtained from the computer manufacturer; this would include items 4 and 6. The Fourier processor manufacturer should provide the educational requirement to satisfy item 8. The remaining educational training will have to be provided by either in-plant or out-of-plant educational facilities. Out-of-plant facilities include local colleges and universities. In addition, there are some organizations whose sole function is to provide computer science education and training[11].

Environmental science journals and organizations have an equal responsibility to provide their readers and members with the opportunity to become knowledgeable in the items listed above. This can be accomplished by inviting papers on appropriate topics fundamental to digital control systems and by organizing and implementing tutorial and workshop lectures at local and national meetings.

G. Costs

1. Hardware Acquisition Costs. Figure 49 illustrates the hardware cost factor range versus the hierarchy of applications defined in Table 7, based upon vendor-supplied information. The basic system (group I, Table 7) has a cost factor of 1.0 units. To add group II capability, it will cost from 2.0 to 2.6 units. That is, if the basic system costs $1 00, the group II system, which is a random system, will cost from $2.00 to 2.60. Having group II capability and desiring group III capability, it will cost approximately an additional $0.03 to 0.05, for a total cost for groups I, II, and III of about $2. 63. The cost function, as defined here, is a cumulative function starting from the required basic control system. No other information is implied from Fig. 49. That is, it is impossible to obtain the cost factor for going from a sine system to a random system from Fig. 49. For example, this figure shows a cost factor of about 0.5 going from a random system (having the basic system) to a sine system. But the author has determined that a cost factor of from 1.1 to 1.2 exists going from a complete n-channel sine system to an n-channel random system.

The important point about the Fig. 49 cumulative cost function is that it starts to flatten out above the group IV application. The steepest part of the curve is between group I and group II. To encompass all the applications in Fig. 49, group VII will require less than twice the money required to get

[11] Two such organizations are the Institute for Advanced Technology (Control Data Corp.), 5272 River Road, Washington, D.C. 20016, and RCA Institutes, 132 West 31st Street, New York, N.Y. 10001.

to group II. The curve flares up at the end (group VII application) because the vendors are presently quoting fixed-head disc prices rather than tape cassette or cartridge prices. Again, the continuous function shown crossing the cost factor ranges does not really exist but is drawn in to illus trate the cost trend as the number of consecutive applications increases.

The cost factors above pertain to hardware only. The total costs must reflect operation, maintenance, and risk, which may outweigh the capital hardware cost over some reasonable time period.

Depreciation (useful life) costs may depend upon the particular application and should be considered in the gross cost figure.

2. System Operating Costs. System operating costs are defined in this paper as five parameters: manpower, maintenance, training, supplies, and power. The most significant factor is manpower; the least significant cost factor is power. If the test system is to be supported by the ideal organization previously discussed, there will be at least two new possible jobs (therefore, two additional people) over and above the personnel requirements for an analog test svstem. Both these new people (system analyst and programmer) could be in the same pay category as a senior engineer.

Cost for power to operate the control system is negligible when considering the overall total operating cost. If the system consumes 4 kW, its cost per hour will be about $0.08, and, based upon normal operating time, the system should cost less than $200/year for electricity, although it could cost up to $700/year if the system is left on all the time, as it probably should.

Maintenance costs (that is, labor to maintain) can be estimated initially by using 1% of the total hardware cost of the system as a monthly figure. This is a reasonable rule of thumb. Maintenance contracts for minicomputers can range from 1% of total hardware cost per month up to 3.5%. Better estimates for new contract maintenance can be made if the vendor is willing to reveal his service call statistics for existing control systems. A computer service call from the manufacturer can be as high as $30/hour plus traveling time.

Training costs can range from $450 for a one- or two-week class, plus transportation and subsistence, to free classes with only transportation and subsistence costs.

Supplies to be considered when estimating operating costs include such items as ink ribbons, TTY paper, printer paper, copy paper, and paper tape. Supplies should also include spare parts, including spare subsystems if required.

H. Customer-Vendor Relations

1. Delivery and Checkout of System. One of the current problems is that, up to now, each delivered computer-controlled test system is in some manner different from the preceding systems, thus making the term "off the shelf" not quite applicable. There is certainly nothing wrong with this. What it means is that the user is specifying a system for his requirements. The vendor is accommodating the user. The degree to which the specified system (both software and hardware) departs from what has previously been delivered will determine the delivery time after purchase.

The first commercially built computer-controlled test system took well over a year to deliver. Test systems now are being delivered in 40-90 days. However, if many "specials" are specified, a longer delivery period should be expected.

Once the system is delivered to the user, a certain amount of time will be required by the vendor to get the system set up and operating. The amount of time required will, again, depend upon the special features and upon the vendor's diagnostic and tester evolution. Assuming that all hardware and software have been proved at the factory, several weeks may be required to get the system set up and operating. The user should realize that large computer systems and data communications systems require months of installation and checkout. The computer-controlled test system is not as complex as these systems. However, the test system is complex enough so that more than one or two days will be required for vendor checkout. The checkout performed by the user may take weeks and will depend upon the prior training and on-site instruction received by the system operator, the system analyst, and the programmer.

2. Division of Responsibility. It is the user's obligation to determine the division of responsibility between himself and the vendor, and it is the vendor's obligation to inform the user of exactly what he, the vendor, is responsible for, prior to consummating the procurement. The division of responsibility must be stated in the procurement specification, and, even then, questions will come up, possibly during the procurement and surely after delivery of the test system. Again, program algorithms and listings fall in this category, to name only one important potential problem. A good deal of thought and effort on the part of the user must go into specifying the system requirements. On the other hand, the vendor is obliged to answer the procurement specification truthfully, listing all the system constraints so that there will be no misunderstanding between the user and the vendor as to what the system can and cannot do when delivered.

ACKNOWLEDGMENT

The authors are indebted to W. Boctor for his preparation of the sections on the Fast Fourier Transform and Sine Servo Control. We wish also to acknowledge the efforts of J. Jorgensen, who drew all of the illustrations, and D. Maple, D. Fulton, and H. Valtier, who furnished editing assistance.

REFERENCES

(1) Chapman, C. P., Shipley, J., and Heizman, C. L., "A Digitally Controlled Vibration or Acoustics Testing System: Parts I, II, III," Proceedings, Mt. Prospect, Ill.,

Institute of Environmental Sciences, 1969, pp. 387-409.

(2) Heizman, Charles L., "A High Performance Vibration Control and Analysis System," Sound and Vibration, July 1973, pp. 8-18.

(3) Anderson, G.C., Finnie, B.W., and Roberts, G.T.; "Pseudo-Random and Random Test Signals," Hewlett-Packard Journal, Sept. 1967, p. 8.

(4) Korn, G.A., Random-Process Simulation and Measurements, McGraw-Hill Book Co., Inc., New York, 1966, p. 81.

(5) Davies, W.D.T., "Random Signal Testing for Evaluation System Dynamics," Wireless World, Aug. 1966, p. 409.

(6) Davies, W.D.T., "Generation and Properties of Maximum-Length Sequences," Control, July 1966, p. 364.

(7) Bendat, J.S., and Piersol, A.G., Measurement and Analysis of Random Data, John Wiley and Sons, Inc., 1966, pp. 4-8.

(8) Reiner, R.L., "A Digitally-Controlled Sweep Oscillator," Proceedings, Mt. Prospect, Ill., Institute of Environmental Sciences, 1970, pp. 52-57.

(9) Bosso, F.C., "Sine-Test Electrodynamic Vibration Systems Using Digital Control," Proceedings, Mt. Prospect, Ill., Institute of Environmental Sciences, 1970, pp. 61-69.

(10) Hosking, Rodger H., "Frequency Synthesizers— Direct, Indirect or Direct Digital?" Electronic Products, Dec. 17, 1973.

(11) Hosking, Rodger H., "Direct Digital Frequency Synthesis," 1973 IEEE Intercon Technical Papers, New Developments in Signal Generators Session.

(12) Smallwood, D.O., "Methods Used to Match Shock Spectra Using Oscillatory Transients," Proceedings, Mt. Prospect, Ill., Institute of Environmental Sciences, 1974, pp. 409-420.

(13) Barthmaier, J.P., "Two Approaches to Computer-Controlled Shock Synthesis," Proceedings, Pittsburgh, Pa., Instrument Society of America, Instrumentation in the Aerospace Industry, Vol 20, Advances in Test Measurement, Vol II, 1974, pp. 313-322.

(14) Morgan, H.C., "Telemetry Course — Part III, Time-Division Multiplexing," The Electronic Engineer, Feb. 1969, pp. 66-73.

(15) Gordon, B.M., "Digital Sampling and Recovery of Analog Signals," Electrical and Electronic Engineering (EEE), May 1970, pp. 65-75.

(16) Bendat, J.S., and Piersol, A.G., "Random Data: Analysis and Measurement Procedures," Wiley-Interscience, New York, pp. 315-320.

(17) Cooley, J.W., and Tukey, J.W., "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematical Computation, Vol. 19, 1965, pp. 297-301.

(18) Sloane, E.A., "Optimum Random Vibration Control Using Transfer Function Analysis," Proceedings, Mt. Prospect, Ill., Institute of Environmental Sciences, 1972, pp. 448-454.

(19) Sotomayor, B.R., "Software Requirements for Computer Control of Random Noise Vibration Tests", Proceedings, Mt. Prospect, Ill., Institute of Environmental Sciences, 1971, pp. 145-155.

(20) Favour, J.D., LeBrun, J.M., and Young, J.P., "Transient Waveform Control of Electromagentic Test Equipment," 40th Shock and Vibration Bulletin, Part 2, December 1969.

(21) Hunter, N.F., Jr., "Transient Waveform Reproduction on Hydraulic Actuators Using a Non-linear Gain Estimation Technique," Institute of Environmental Sciences 1974 Proceedings, pp. 202-206, Institute of Environmental Sciences, Mt. Prospect, Ill.

(22) Kim, B.K., "Induced Shock Pulse, Testing by Transient Waveform Control," Quarterly Technical Review, Vol. 3, No. 2, Jet Propulsion Laboratory, Pasadena, Calif., July 1973.

(23) Yang, R.C., and Saffell, H.R., "Development of a Waveform Synthesis Technique — A Supplement to Response Spectrum as a Definition of Shock Environment," 42nd Shock and Vibration Bulletin, Part 2, January 1972, pp. 45-54.

(24) Chapman, C.P., "Computer-Controlled Environmental Test Systems: Criteria for Selection, Installation, and Maintenance," SAE Paper No. 720819, 1972.

(25) Chapman, C.P. and Sotomayor, B.R., "Computer Controlled Test System for Random Noise Induced Vibration and Acoustics Environments." Test Engineering and Management, Vol. XXV, Nos. 2 and 3, Feb.-Mar. 1971, p. 15.

(26) Curtis, A.J., "On the Performance of TDM Averagers in Random Vibration Tests." 42nd Shock and Vibration Bulletin, Part 3, Jan. 1972.

(27) Chapman, C.P., 66 Scanning Techniques for Random Noise Testing." J. Environ. Sci., Vol. 11, No. 1, Feb. 1968, pp. 27-35.

(28) Norin, Robert S., "A Multi-Channel, Multi-strategy Sinusoidal Vibration Control System," Proceedings, Mt. Prospect, Ill., Institute of Environmental Sciences, 1974, pp. 222-228.

(29) Chapman, C.P., and Sotomayor, B.R., "Off-Line Data Analysis and Data Reduction from Digitally Controlled Random Test Systems," Proceedings, Mt. Prospect, Ill., Institute of Environmental Sciences, 1972, pp. 120-137.

(30) Goode, H.H., and Machol, R.E., *System Engineering*, McGraw-Hill, New York, 1957.

(31) Machol, R.E., Tanner, W.P., and Alexander, S., *System Engineering Handbook*, McGraw-Hill, New York, 1965.

(32) Wagner, H.M., *Principles of Operations Research with Applications to Managerial Decisions*, Prentice-Hall, New York, 1969.

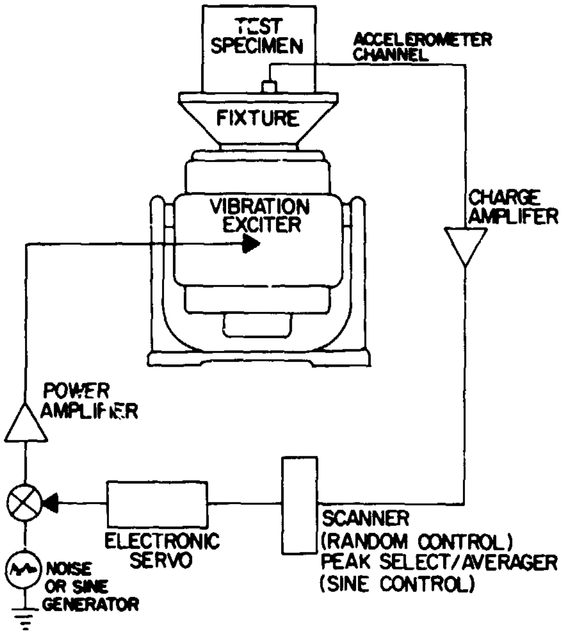


Fig. 1. Analog vibration test system

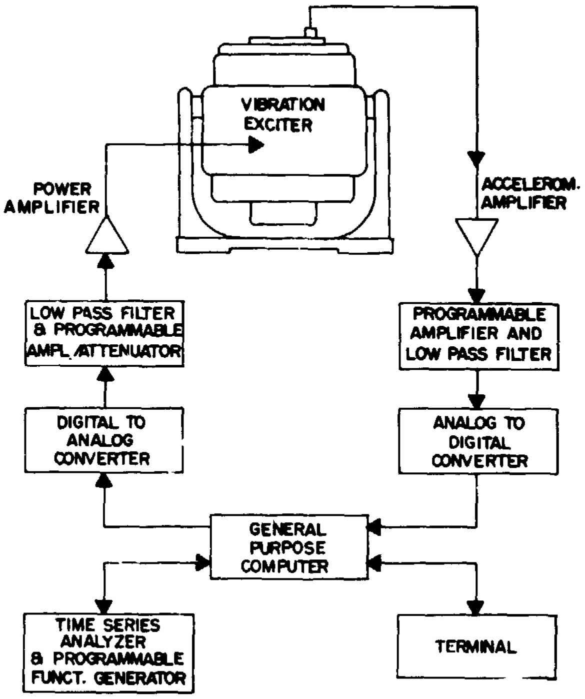

VIBRATION EXCITER
POWER AMPLIFIER
ACCELEROM. AMPLIFIER
LOW PASS FILTER & PROGRAMMABLE AMPL./ATTENUATOR
PROGRAMMABLE AMPLIFIER AND LOW PASS FILTER
DIGITAL TO ANALOG CONVERTER
ANALOG TO DIGITAL CONVERTER
GENERAL PURPOSE COMPUTER
TIME SERIES ANALYZER & PROGRAMMABLE FUNCT. GENERATOR
TERMINAL

Fig. 2. Computer controlled vibration test system

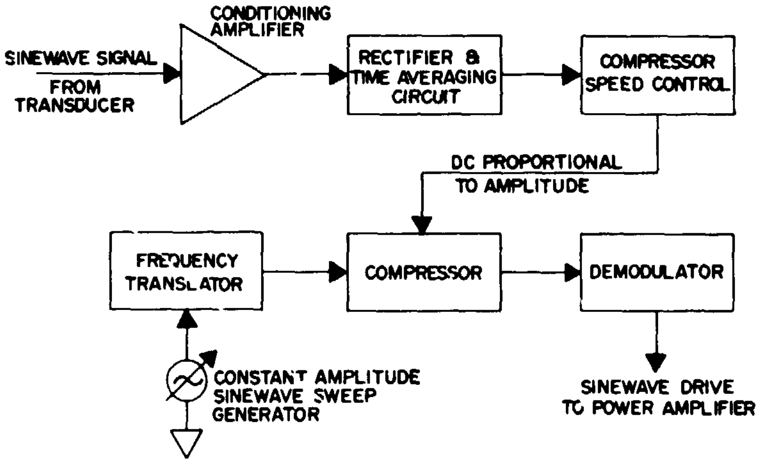

CONDITIONING AMPLIFIER
SINEWAVE SIGNAL FROM TRANSDUCER
RECTIFIER & TIME AVERAGING CIRCUIT
COMPRESSOR SPEED CONTROL
DC PROPORTIONAL TO AMPLITUDE
FREQUENCY TRANSLATOR
COMPRESSOR
DEMODULATOR
CONSTANT AMPLITUDE SINEWAVE SWEEP GENERATOR
SINEWAVE DRIVE TO POWER AMPLIFIER

Fig. 3. Analog sine servo (conceptual)

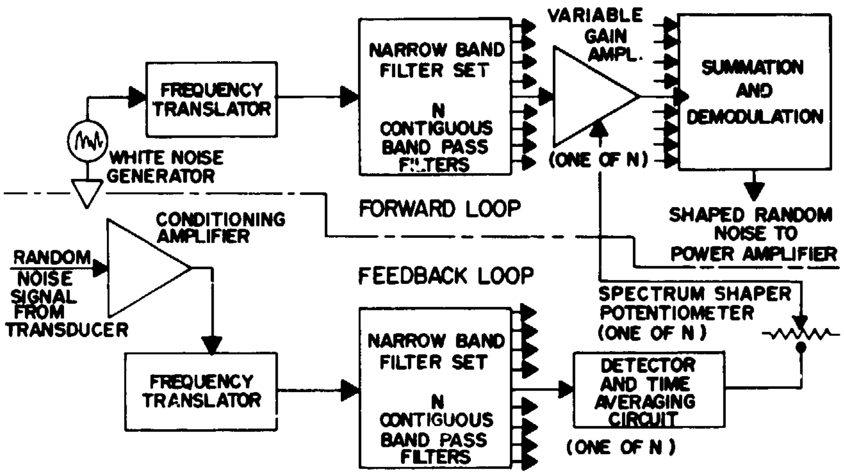

FREQUENCY TRANSLATOR
NARROW BAND FILTER SET
N CONTIGUOUS BAND PASS FILTERS
VARIABLE GAIN AMPL.
SUMMATION AND DEMODULATION
WHITE NOISE GENERATOR
(ONE OF N)
FORWARD LOOP
SHAPED RANDOM NOISE TO POWER AMPLIFIER
CONDITIONING AMPLIFIER
RANDOM NOISE SIGNAL FROM TRANSDUCER
FEEDBACK LOOP
SPECTRUM SHAPER POTENTIOMETER (ONE OF N)
FREQUENCY TRANSLATOR
NARROW BAND FILTER SET
N CONTIGUOUS BAND PASS FILTERS
DETECTOR AND TIME AVERAGING CIRCUIT
(ONE OF N)

Fig. 4. Analog random noise servo (conceptual)

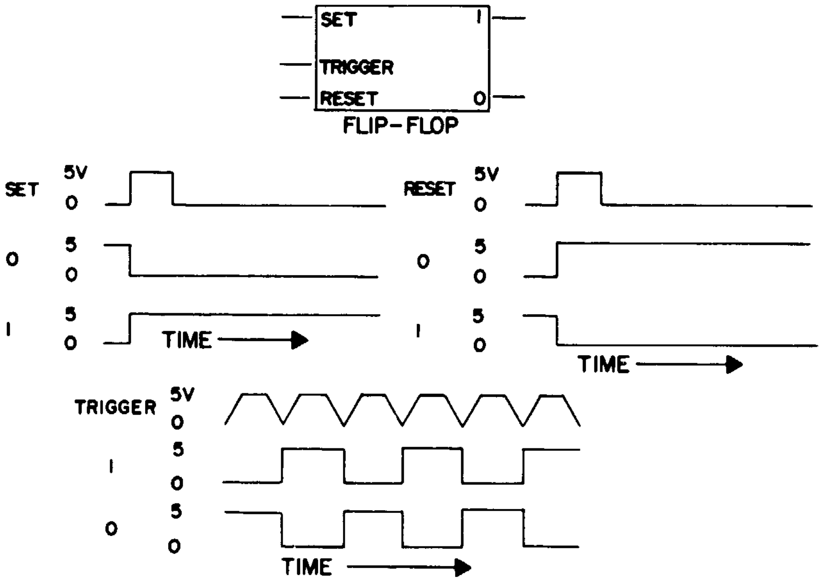


Fig. 5. Input/output flip-flop characteristics

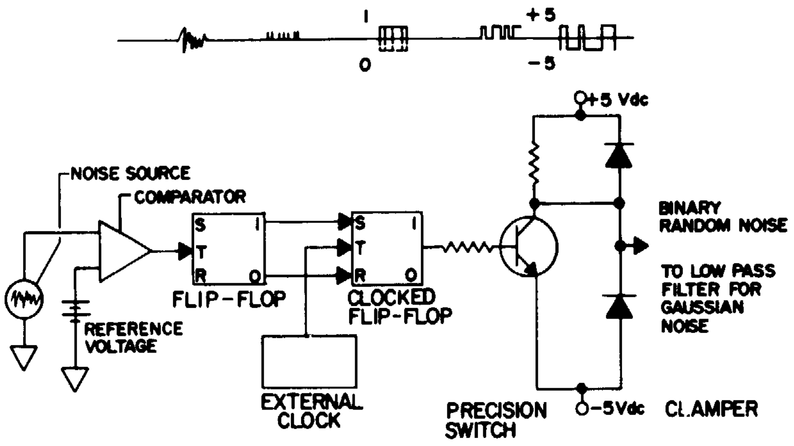
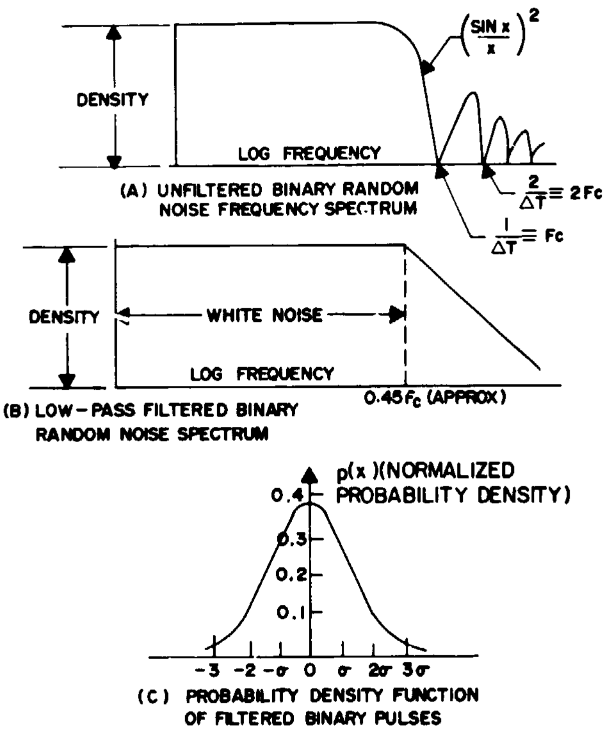


Fig. 6. Binary random noise generator (conceptual)



(A) UNFILTERED BINARY RANDOM NOISE FREQUENCY SPECTRUM

(B) LOW-PASS FILTERED BINARY RANDOM NOISE SPECTRUM

(C) PROBABILITY DENSITY FUNCTION OF FILTERED BINARY PULSES

Fig. 7. Spectrum and amplitude distribution characteristics of binary random noise

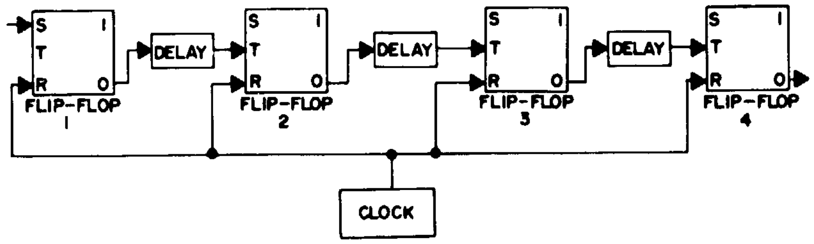
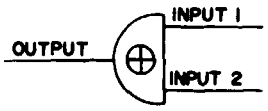


Fig. 8. Four-stage shift register (conceptual)



(a) SYMBOL

INPUT		OUTPUT
1	2	
0	0	0
0	1	1
1	0	1
1	1	0

(b) TRUTH TABLE

Fig. 9. Exclusive-OR gate

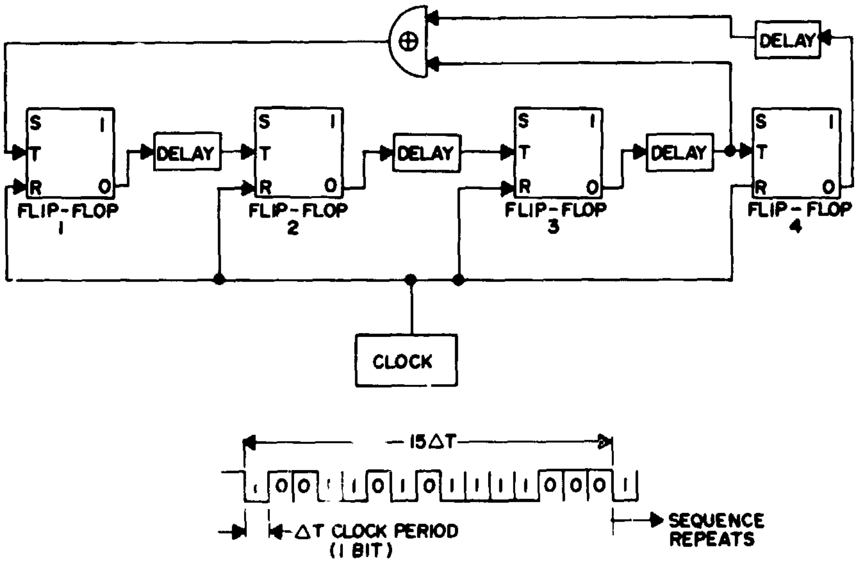


Fig. 10. Maximum-length sequence generation

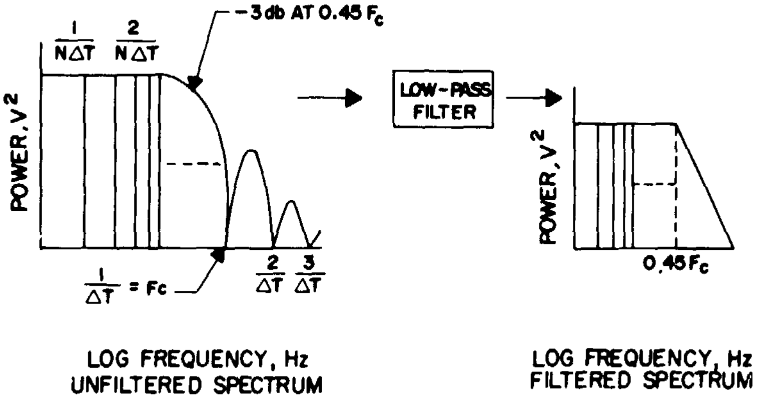
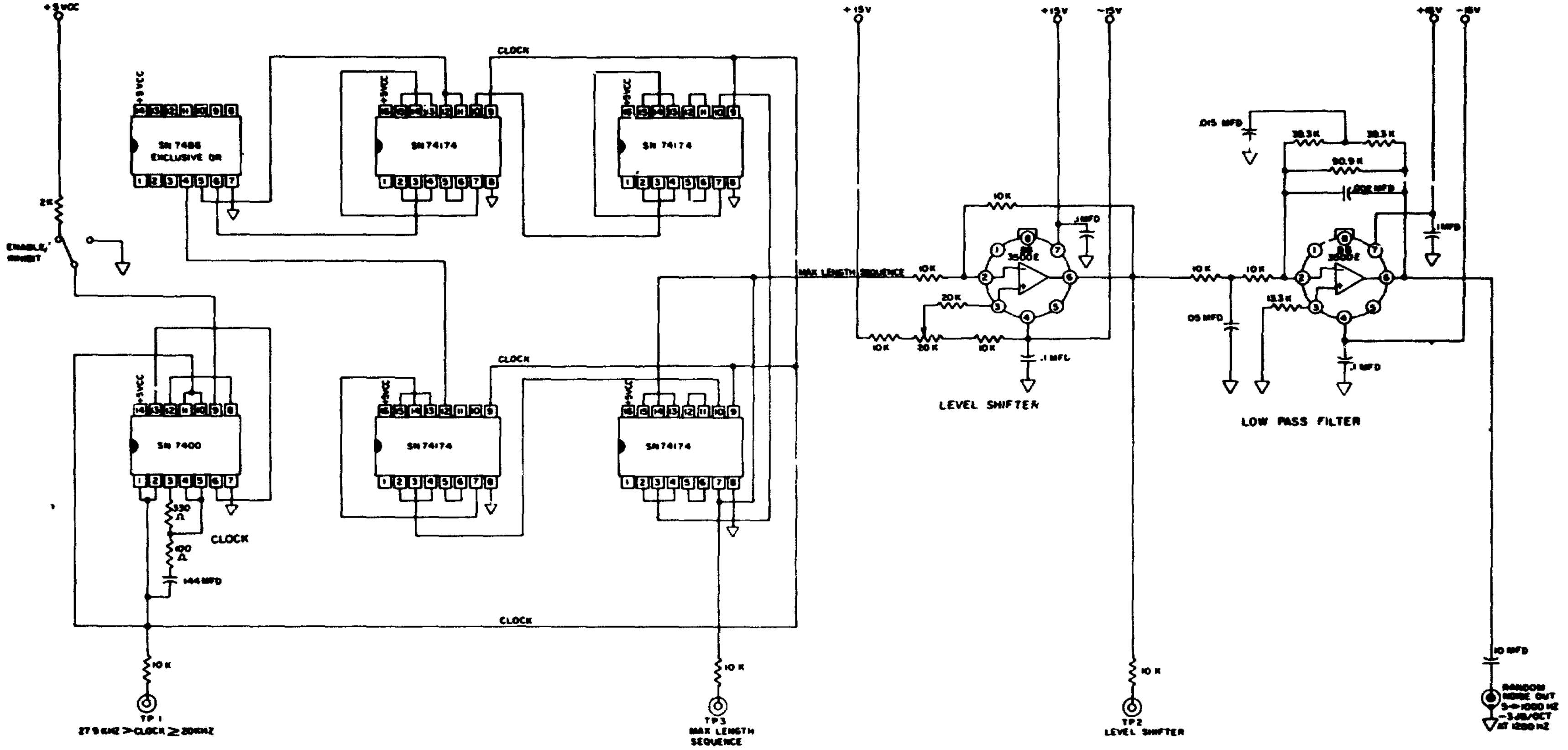


Fig. 11. Pseudorandom noise spectra characteristics



1 ALL IC'S ARE TOP VIEWS

Fig. 12. Pseudorandom noise generator, schematic diagram

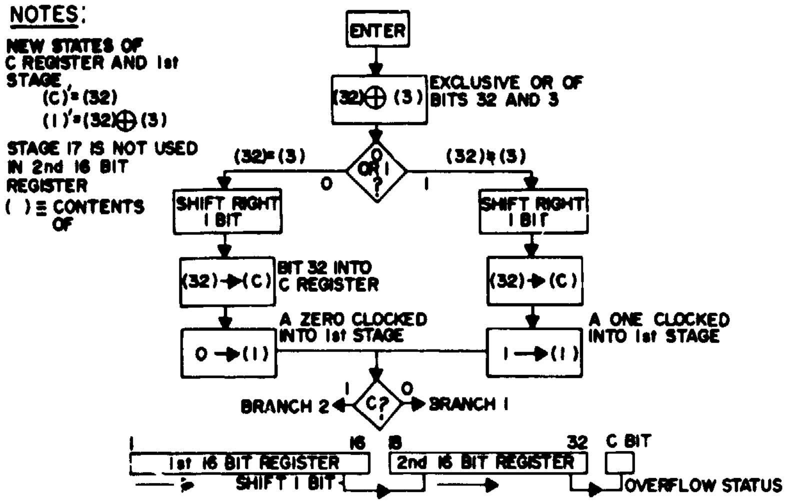


Fig. 13. Computer register hardware configuration and software flow chart requirements to generate a maximum length pseudorandom sequence

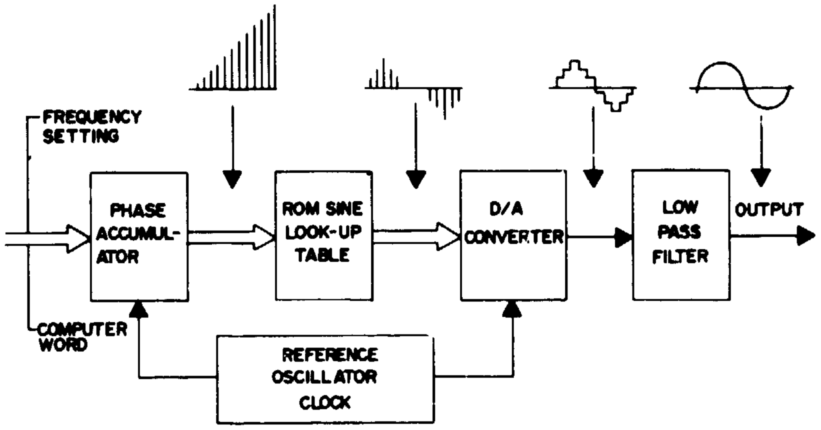


Fig. 14. Direct digital frequency synthesizer (conceptual)

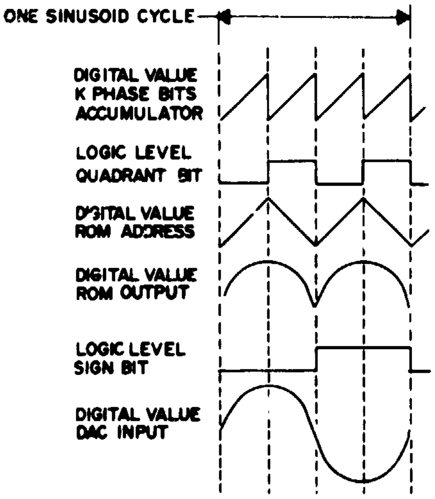


Fig. 15. Accumulator and ROM signals

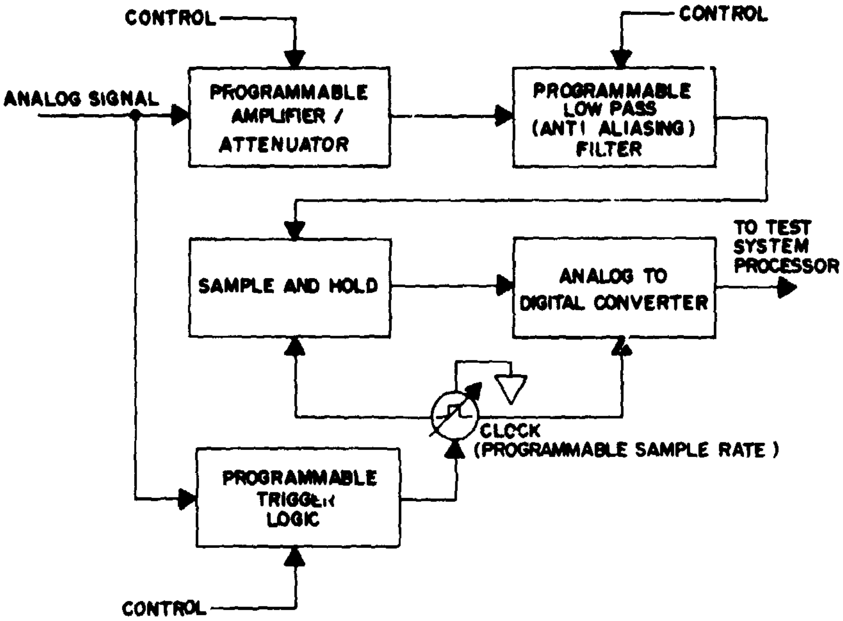

CONTROL
CONTROL
ANALOG SIGNAL
PROGRAMMABLE AMPLIFIER / ATTENUATOR
PROGRAMMABLE LOW PASS (ANTI ALIASING) FILTER
SAMPLE AND HOLD
ANALOG TO DIGITAL CONVERTER
TO TEST SYSTEM PROCESSOR
CLOCK (PROGRAMMABLE SAMPLE RATE)
PROGRAMMABLE TRIGGER LOGIC
CONTROL

Fig. 16. Data channel of input signal conditioning subsystem (conceptual)

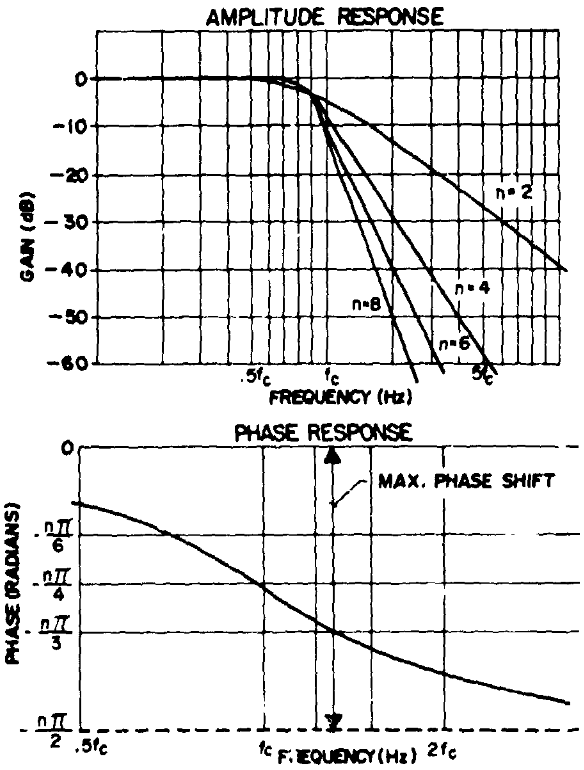


Fig. 17. Butterworth filter transfer function

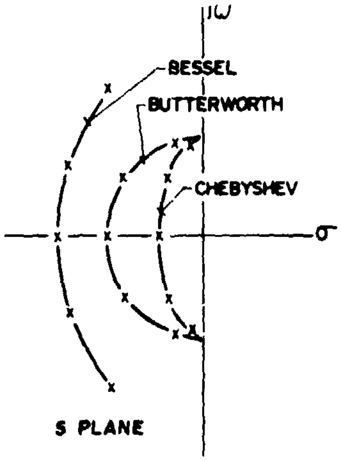
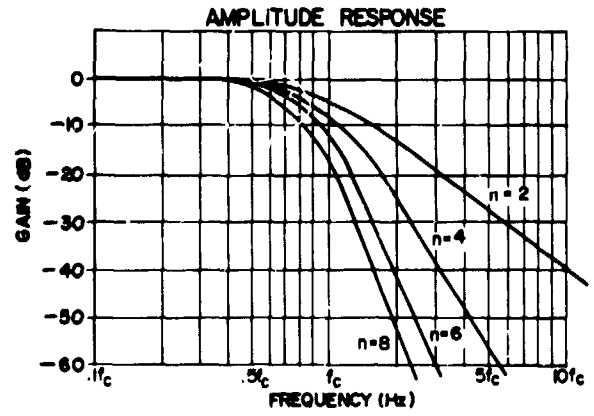


Fig. 18. S plane filter pole locations



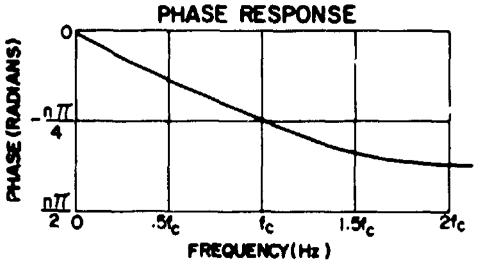


Fig. 19. Bessel filter transfer function

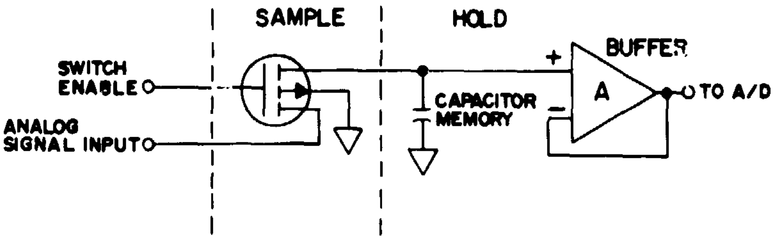


Fig. 20. MOS FET sample-and-hold

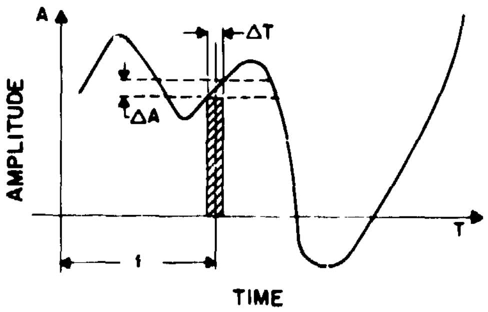


Fig. 21. Effect of finite conversion (measurement) time on single point sample (Ref. 15)

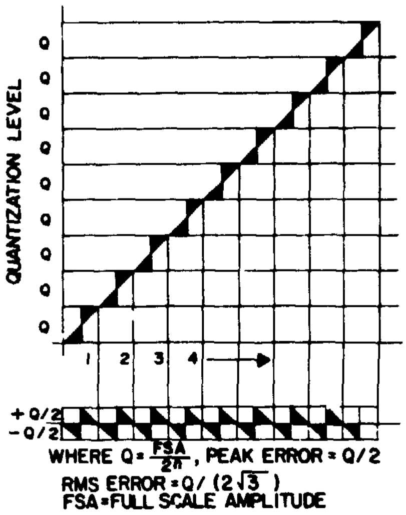

QUANTIZATION LEVEL
Q
Q
Q
Q
Q
Q
Q
Q
Q
1
2
3
4
+Q/2
-Q/2
WHERE Q = FSA/2n, PEAK ERROR = Q/2
RMS ERROR = Q/(2√3)
FSA = FULL SCALE AMPLITUDE

Fig. 22. Quantization effect and the resulting error function (Ref. 15)

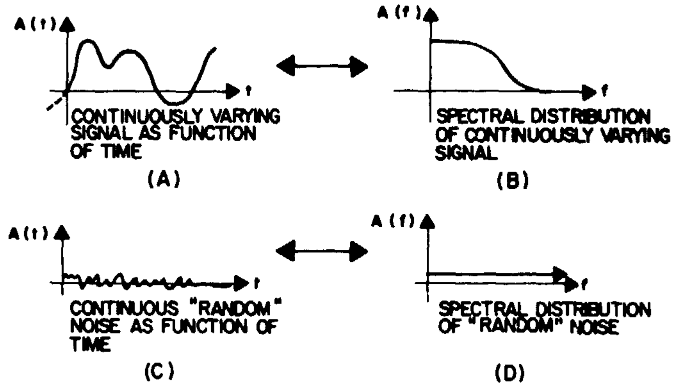

A(t)
t
CONTINUOUSLY VARYING SIGNAL AS FUNCTION OF TIME
(A)
A(f)
f
SPECTRAL DISTRIBUTION OF CONTINUOUSLY VARYING SIGNAL
(B)
A(t)
t
CONTINUOUS "RANDOM" NOISE AS FUNCTION OF TIME
(C)
A(f)
f
SPECTRAL DISTRIBUTION OF "RANDOM" NOISE
(D)

Fig. 23. Continuously varying signal (a) and (b) and random noise (c) and (d) displayed as amplitude function of time (left) and as frequency spectra (right) (Ref. 15)

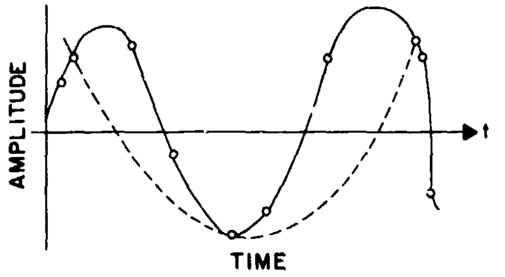


Fig. 24. Effect of sampling at inadequate rate and, thus, creating "aliases" (Ref. 15)

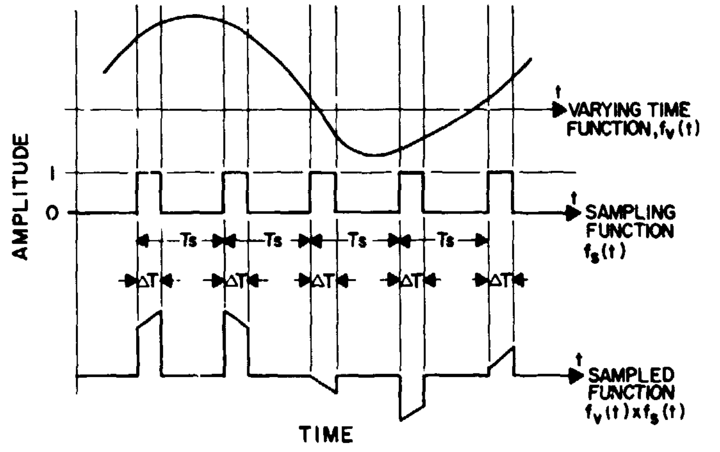

AMPLITUDE
t
VARYING TIME FUNCTION, $f_v(t)$
1
0
t
SAMPLING FUNCTION $f_s(t)$
Ts
Ts
Ts
Ts
ΔT
ΔT
ΔT
ΔT
ΔT
t
SAMPLED FUNCTION $f_v(t) \times f_s(t)$
TIME

Fig. 25. Effect of multiplying continuously varying signal by flat-topped sampling function (Ref. 15)

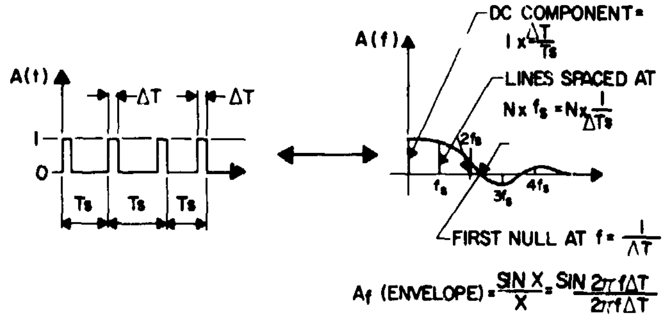

A(t)
ΔT
ΔT
1
0
Ts
Ts
Ts
A(f)
DC COMPONENT = $1 \times \frac{\Delta T}{Ts}$
LINES SPACED AT $N \times f_s = N \times \frac{1}{\Delta Ts}$
$2f_s$
f_s
$3f_s$
$4f_s$
FIRST NULL AT $f = \frac{1}{\Delta T}$
A_f (ENVELOPE) $= \frac{\text{SIN } X}{X} = \frac{\text{SIN } 2\pi f \Delta T}{2\pi f \Delta T}$

Fig. 26. Relationship between time function and frequency function for flat-topped repetitive pulses (Ref. 15)

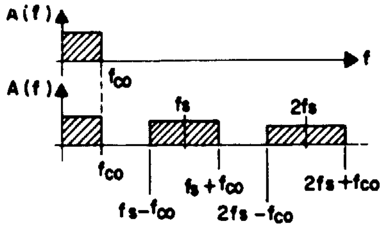


Fig. 27. Spectrum that results from multiplying a limited bandwidth signal (characterized by an ideal filter) by a sampling signal at rate f_s (Ref. 15)

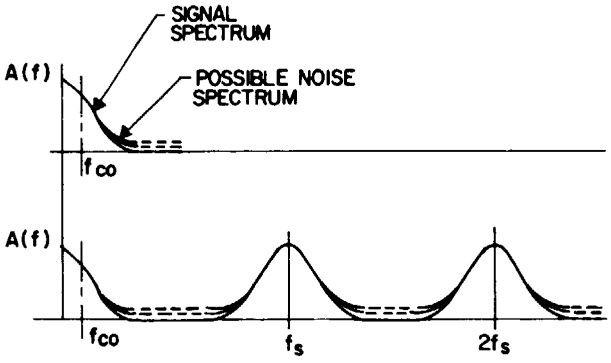


Fig. 28. Spectrum that results from multiplying a realistic signal (that is noisy and is characterized by a real filter) by a sampling signal of rate f_s (Ref. 15)

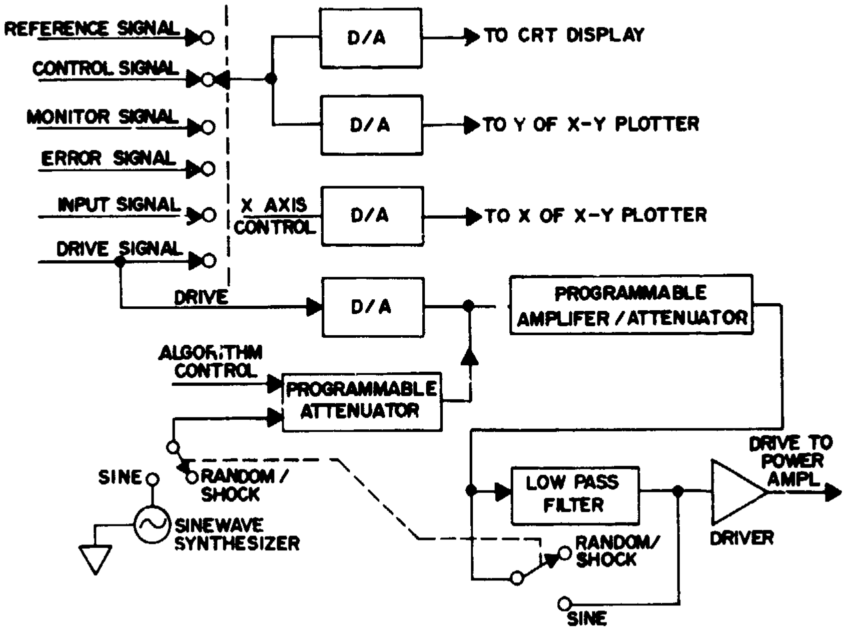


Fig. 29. D/A conversion requirements, digital vibration control

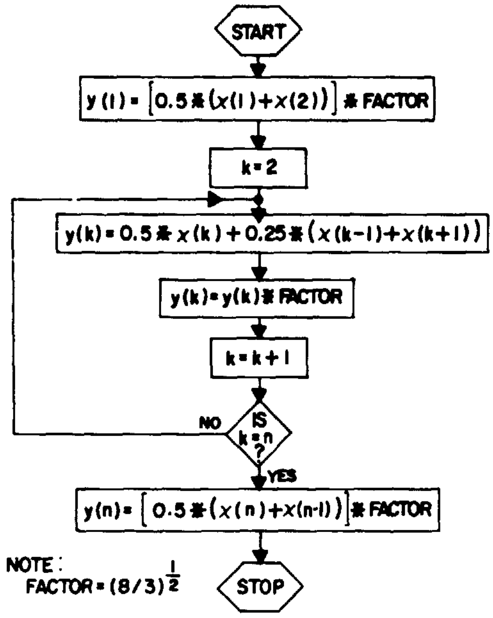

START
y(1) = [0.5*(x(1)+x(2))] * FACTOR
k=2
y(k) = 0.5*x(k) + 0.25*(x(k-1)+x(k+1))
y(k)=y(k)* FACTOR
k=k+1
NO
IS k=n ?
YES
y(n) = [0.5*(x(n)+x(n-1))] * FACTOR
NOTE:
FACTOR = (8/3)^1/2
STOP

Fig. 30. Hanning data weighting (normalized)

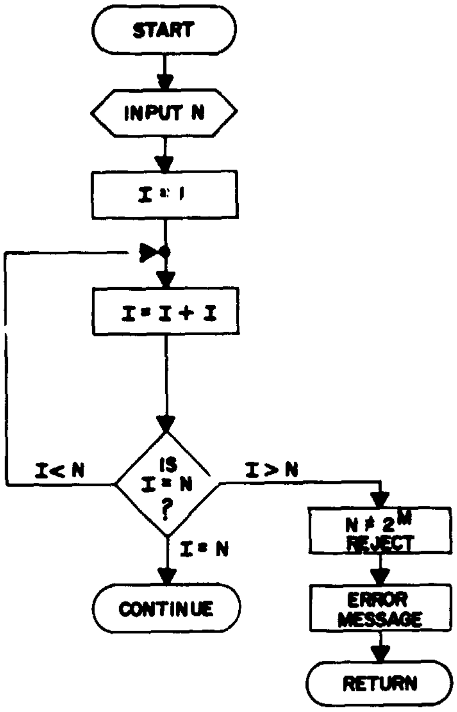


Fig. 31. Flowchart for power-of-two frame size test

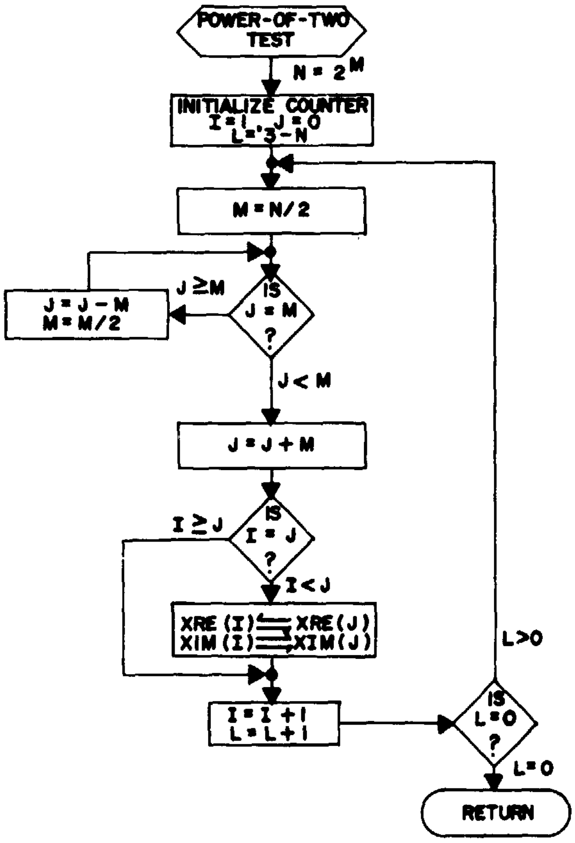


Fig. 32. Flowchart for input shuffling process

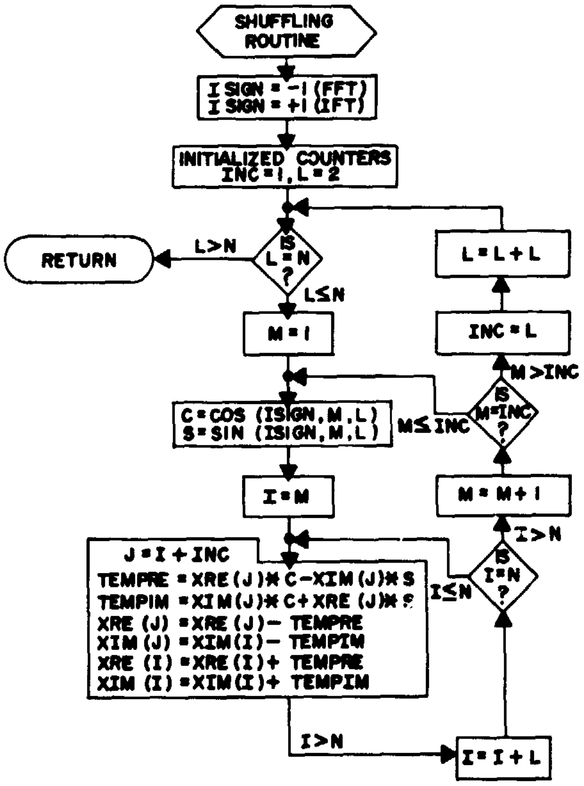


Fig. 33. Flowchart FFT/IFT algorithm

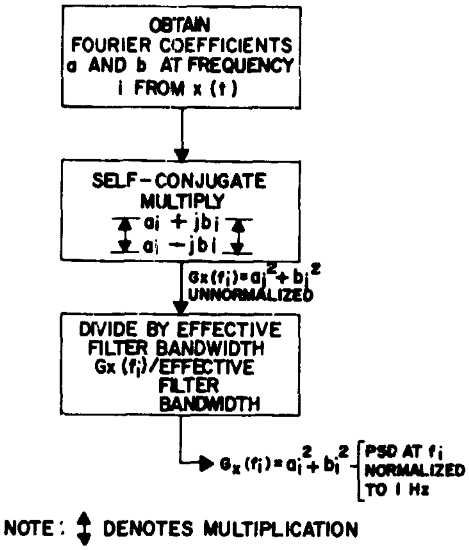

OBTAIN FOURIER COEFFICIENTS a AND b AT FREQUENCY i FROM x (t)
SELF-CONJUGATE MULTIPLY
ai + jbi
ai − jbi
Gx(fi) = ai² + bi² UNNORMALIZED
DIVIDE BY EFFECTIVE FILTER BANDWIDTH Gx (fi)/EFFECTIVE FILTER BANDWIDTH
Gx (fi) = ai² + bi²
PSD AT fi NORMALIZED TO 1 Hz
NOTE: ↕ DENOTES MULTIPLICATION

Fig. 34. Power spectral density process

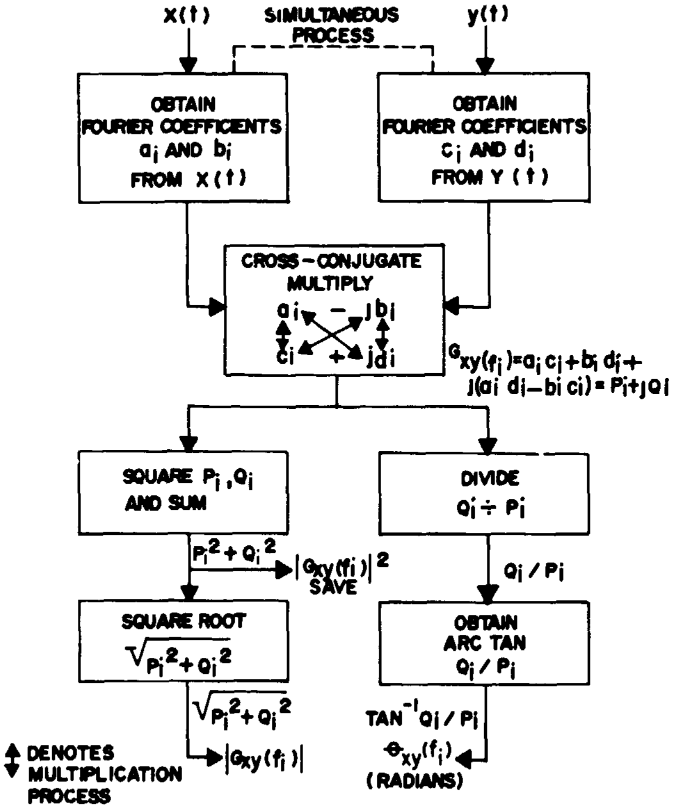


Fig. 35. Cross spectrum process

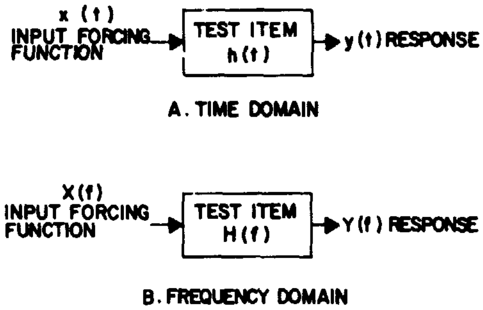


Fig. 36. Input/output signal concepts

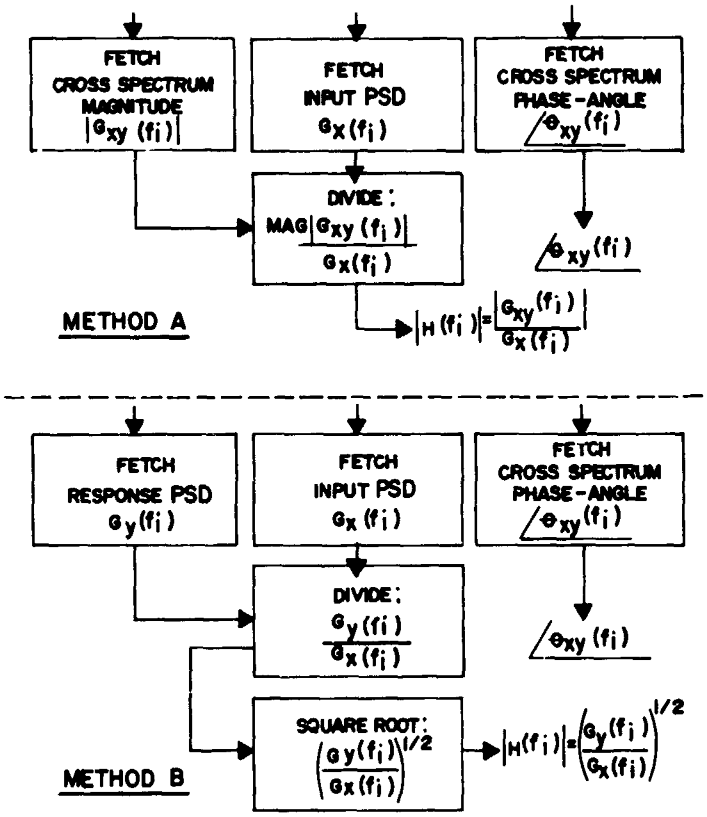

FETCH CROSS SPECTRUM MAGNITUDE $|G_{xy}(f_i)|$
FETCH INPUT PSD $G_x(f_i)$
FETCH CROSS SPECTRUM PHASE-ANGLE $\angle\theta_{xy}(f_i)$
DIVIDE: MAG $\frac{|G_{xy}(f_i)|}{G_x(f_i)}$
$\angle\theta_{xy}(f_i)$
METHOD A
$|H(f_i)| = \left|\frac{G_{xy}(f_i)}{G_x(f_i)}\right|$
FETCH RESPONSE PSD $G_y(f_i)$
FETCH INPUT PSD $G_x(f_i)$
FETCH CROSS SPECTRUM PHASE-ANGLE $\angle\theta_{xy}(f_i)$
DIVIDE: $\frac{G_y(f_i)}{G_x(f_i)}$
$\angle\theta_{xy}(f_i)$
SQUARE ROOT: $\left(\frac{G_y(f_i)}{G_x(f_i)}\right)^{1/2}$
$|H(f_i)| = \left(\frac{G_y(f_i)}{G_x(f_i)}\right)^{1/2}$
METHOD B

Fig. 37. Frequency response processes

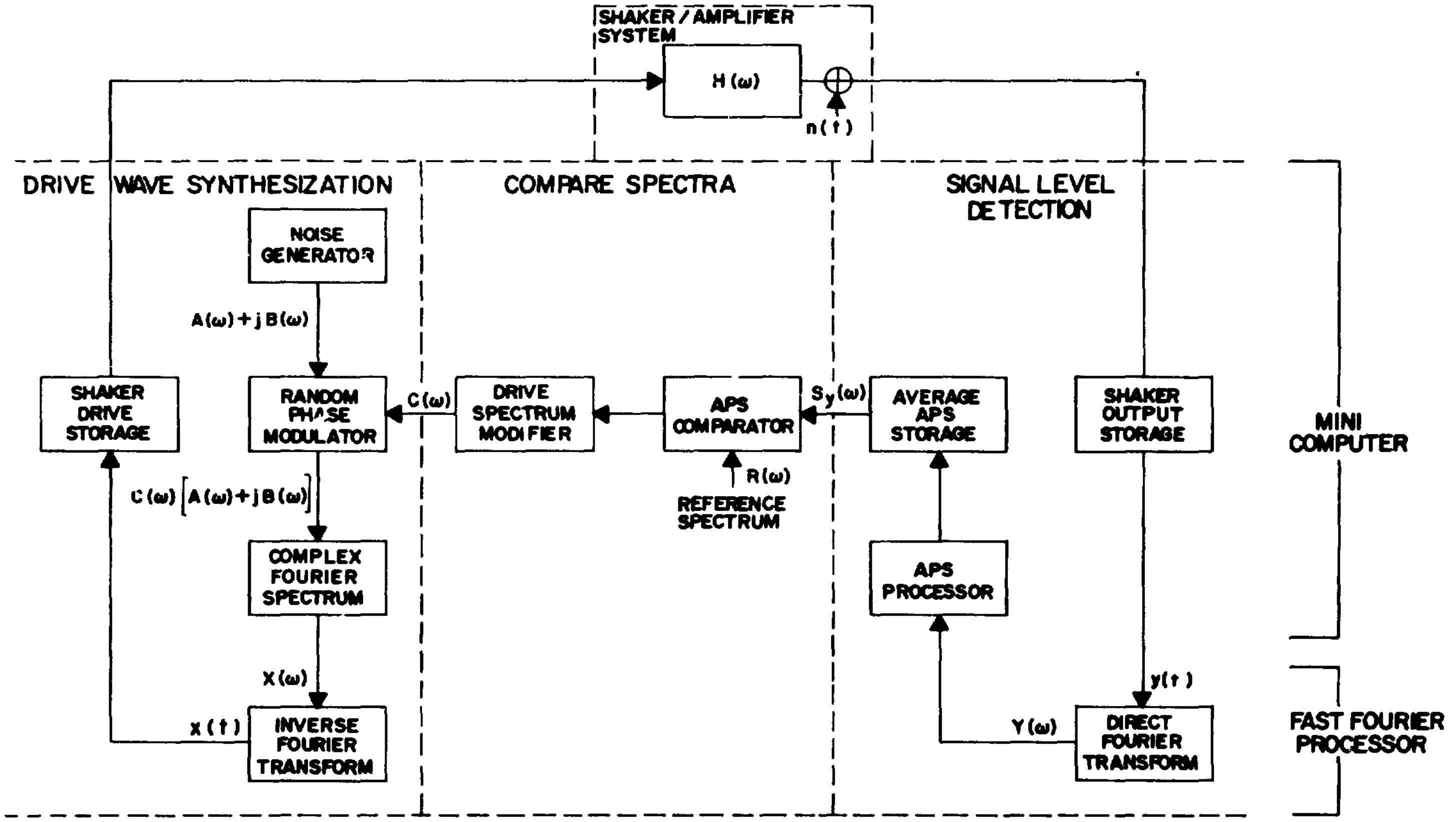


Fig. 38. Random servo control (APS method) functional block diagram

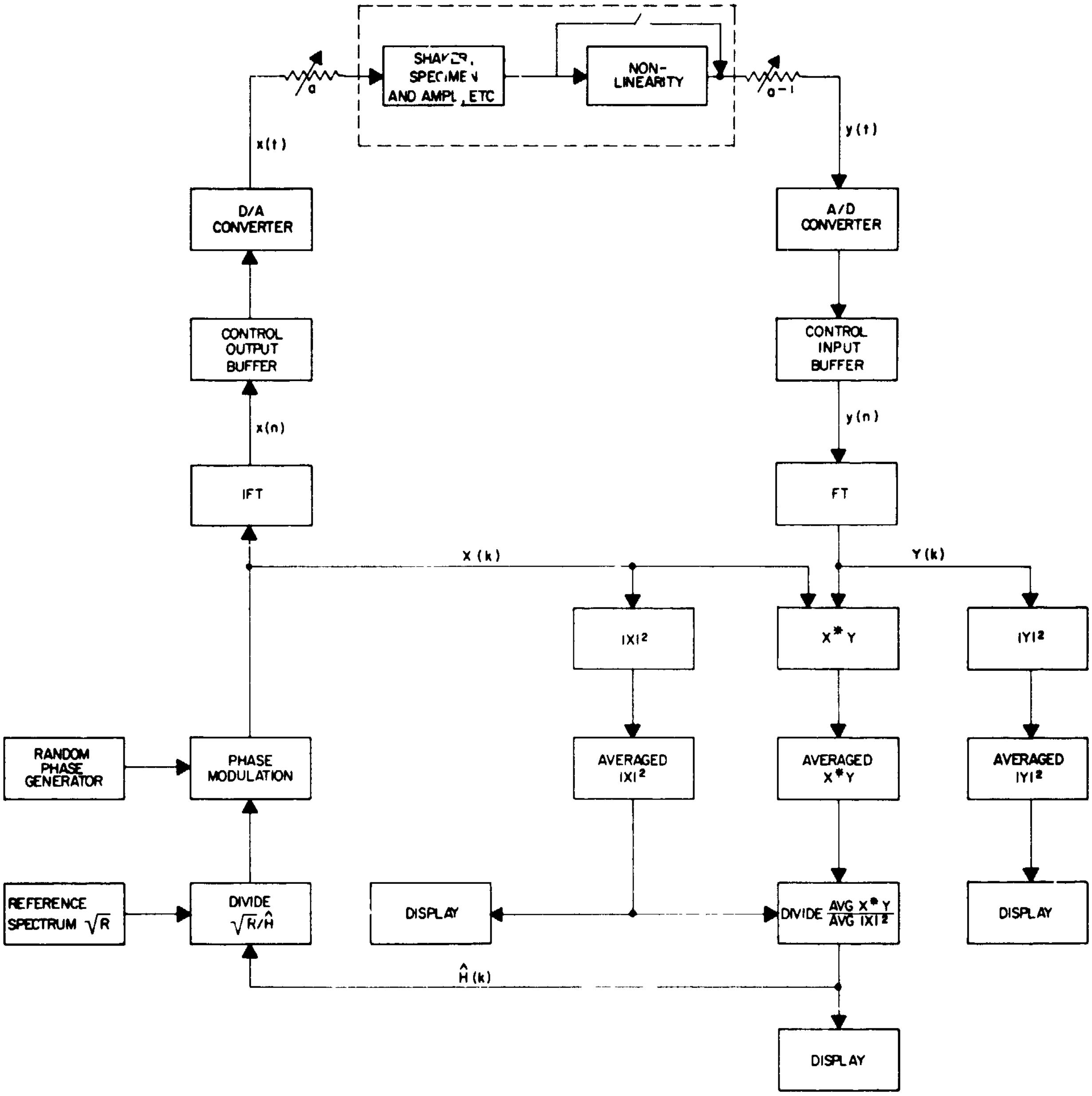


Fig. 39. Functional block diagram of random vibration complex transfer function control method

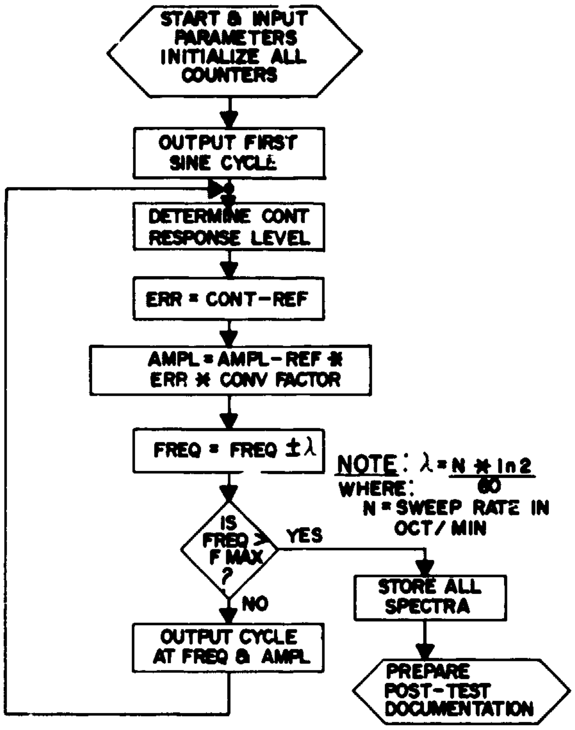

START & INPUT PARAMETERS INITIALIZE ALL COUNTERS
OUTPUT FIRST SINE CYCLE
DETERMINE CONT RESPONSE LEVEL
ERR = CONT-REF
AMPL = AMPL-REF * ERR * CONV FACTOR
FREQ = FREQ ±λ
NOTE: λ = N * ln2 / 60
WHERE:
N = SWEEP RATE IN OCT/MIN
IS FREQ > F MAX ?
YES
NO
STORE ALL SPECTRA
PREPARE POST-TEST DOCUMENTATION
OUTPUT CYCLE AT FREQ & AMPL

Fig. 40. Amplitude servo algorithm for sine-sweep test

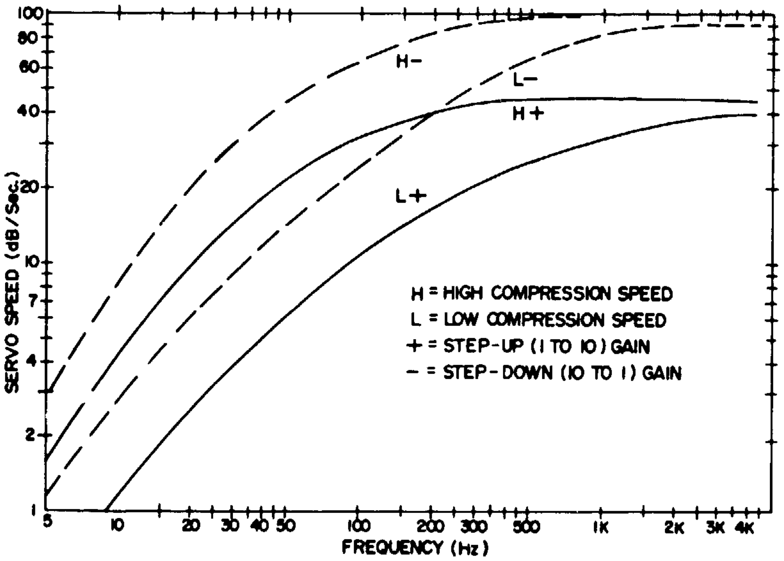


Fig. 41. Servo speed as a function of frequency for typical digital system.

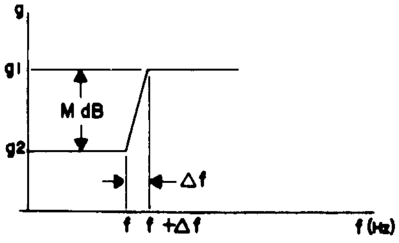


Fig. 42. Relationship between a step jump and the frequency separation necessary for proper servo control

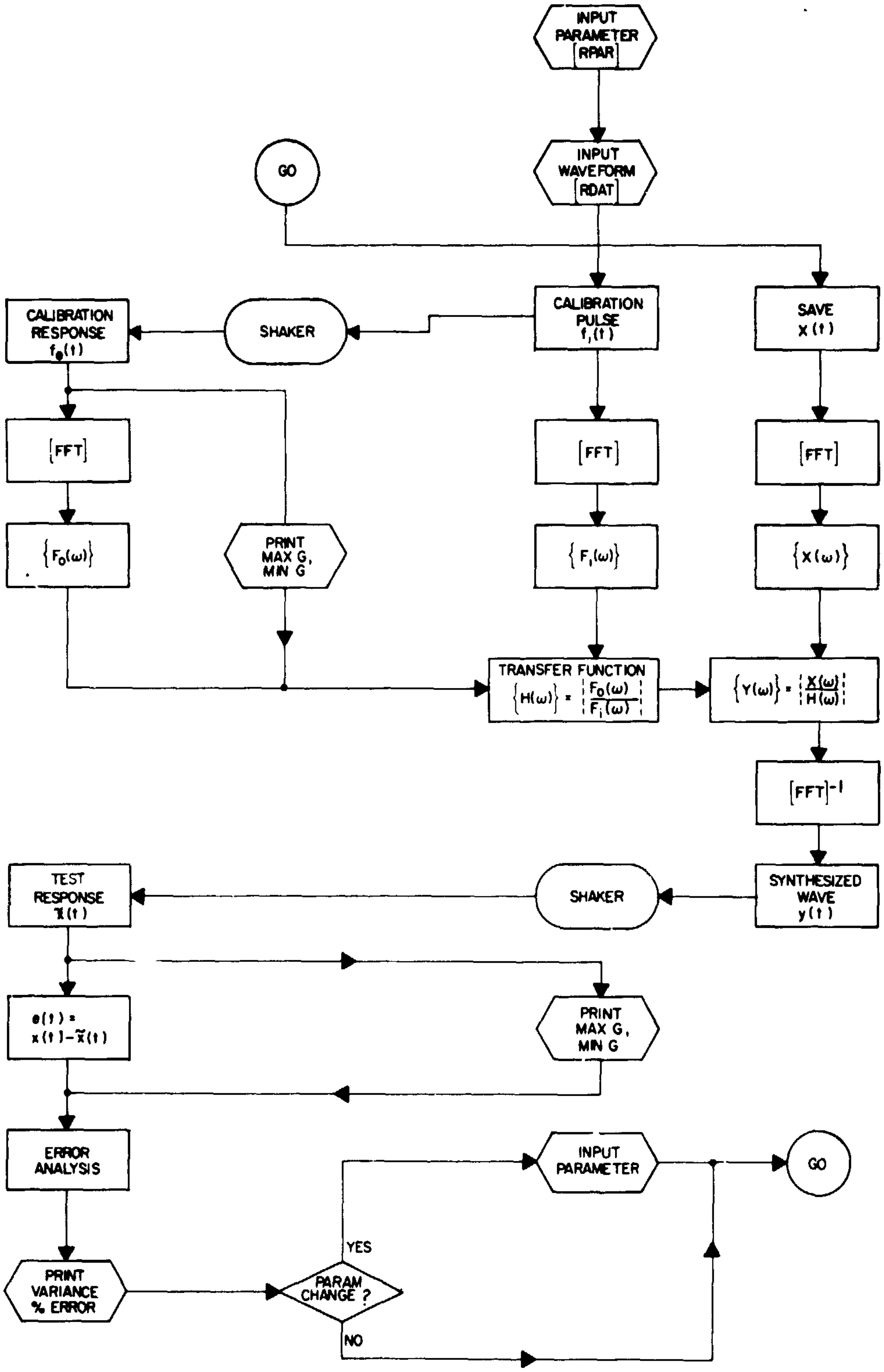


Fig. 43. Software flow chart to synthesize a transient time history

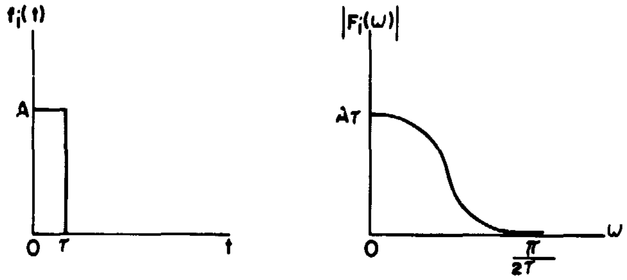


Fig. 44. Transform pair of the TWC initial calibration pulse

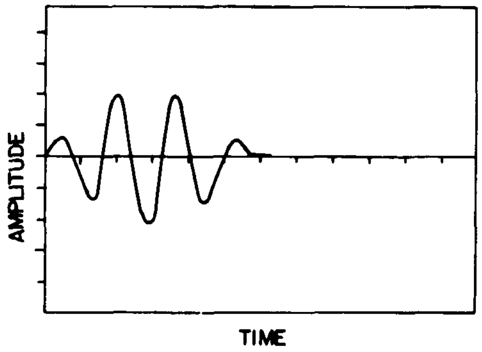


Fig. 45a. Wavelet with 7 half cycles and 60 Hertz fundamental frequency

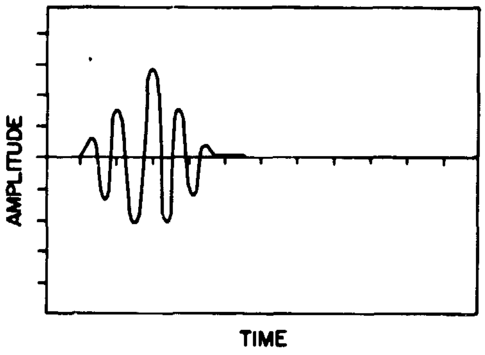


Fig. 45b. Wavelet with 9 half cycles and 120 Hertz fundamental frequency, and a delay of 10 milliseconds

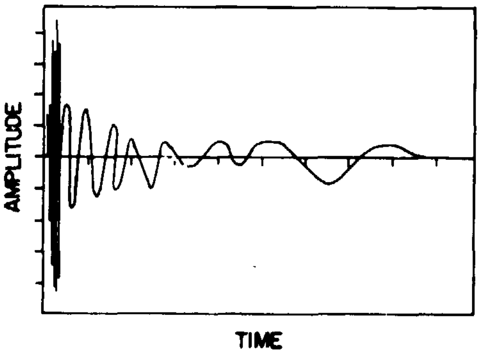


Fig. 46. Composite waveform from typical shock spectrum specification

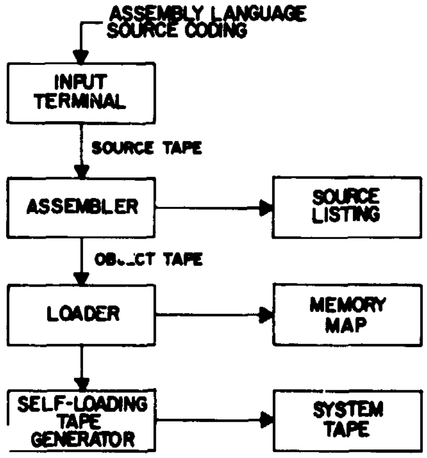


Fig. 47. System program generator procedure

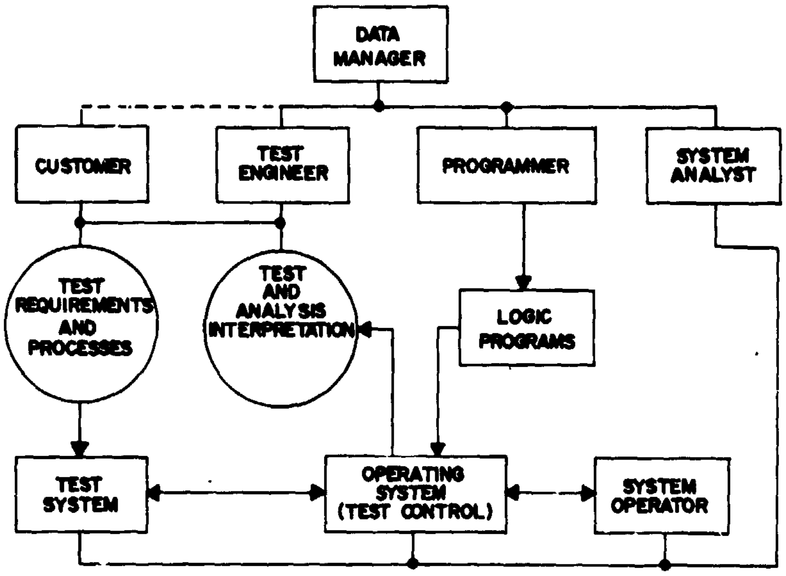


Fig. 48. Operating system and personnel requirements

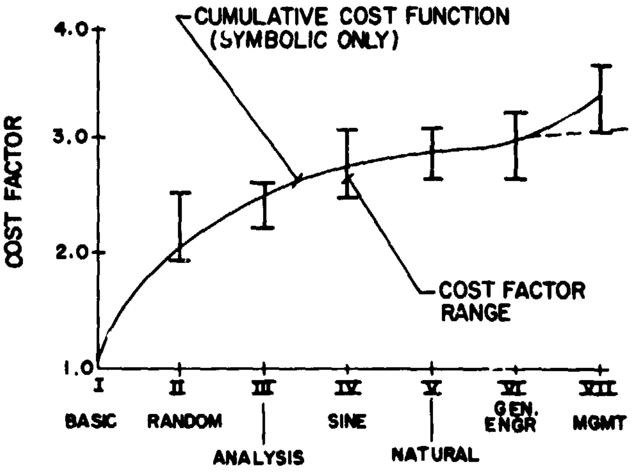


Fig. 49. Cost factor range versus hierarchy of application

APPENDIX

COMPUTERS FOR ENVIRONMENTAL TEST SYSTEMS[12]

I. COMPUTER TECHNOLOGY

Modern-day digital computation started in the mid-1940's with the programmable vacuum tube computers developed for the military for purposes of computing ballistic firing tables. Figure A-1 indicates the time chart of computer technology in the United States. Note that computers for environmental test control were utilized starting in 1968.

Computer technology today is doubling perhaps every four years. Computer hardware costs are being reduced by an order of magnitude every six years. These costs are linked to technology and technological advances associated with batch-fabricated microcircuits. However, peripheral hardware, i. e., that hardware hung onto the computers such as terminals, tape drives, printers, etc., is not dropping in price as rapidly as the computer costs. This is because these peripheral devices are electromechanical in nature. Software costs, on the other hand, are increasing in that software development is tied directly with people's salaries.

It can be seen from Fig. A-1 that the advent of the minicomputer took place about 1964. With the growth of the minicomputer market and the heavy inroads the minicomputer has made into applications areas, the mini is rapidly replacing many of the larger, more costly machines, despite the fact that the applications software is generally a more expensive development, and that the mini, by definition, cannot compete in terms of word and memory size with larger, more powerful computers.

II. COMPUTER APPLICATIONS

Figure A-2 shows the development of computer applications as computer technology advanced. To the application of computing ballistic trajectories was added the computation tasks of simple batch processing of business and scientific jobs. In turn, applications utilizing these machines were tried and developed at a rather rapid rate as the technology advanced even further. Operating systems were configured to implement sequential batch processing of data, asynchronous multiprogramming requirements, and input/output oriented operations. Additional applications stemmed from these applications in turn. The following application areas for present day computer systems are briefly defined:

Process Control. This application includes machine tool control, production line control, oil and gas field monitoring, biomedical monitoring, and laboratory monitoring.
Scientific Computing. This includes equation solving, engineering and system design, system analysis and simulation, and system modeling.
Business Data Processing. Here payrolls are generated, accounts payable are paid, accounts receivable are recorded, inventories are controlled, and in summary, the computer is utilized as a valuable management tool. This includes the very powerful techniques of linear programming, probability theory, reliability, nonlinear programming, statistics, decision theory, queue and game theory, and information theory.
Military/Spacecraft Command and Control. These applications include guidance and tracking, weapons control, navigation, communications control, and cryptographic processing.
Real Time Processing. This includes environmental test control, which is a subset of command and control applications, brokerage systems, and airline reservation systems, to name a few.
Time Sharing. These applications include remote inquiry, scientific computing, and data base access.
Data Communications Control. This is probably the newest application field, which includes message switching, telemetry, communications concentrators, and, most importantly, front ends for the "super computers".
Preprocessing. This application includes buffering, peripheral control, display control, and communications control.

The point here is that computer utilization for environmental test control is only a very limited application in the total hierarchy of applications to date. And still, the computer used for test control can and should certainly be utilized for other than real time processing within the test lab.

III. THE SPECTRUM OF POWER

Computers can be categorized in terms of CPU (Central Processing Unit) capability, word length, memory capacity, input/output (I/O) capability, complexity of interfacing, and supplied software. Cost is another factor. Figure A-3 defines the spectrum of power of computers as a function of word size and cost from the mini to the super number crunchers. The mini is, of course, the most popular type of computer found in the environmental labs these days, although not exclusively.

Minis and other computers are characterized by I/O throughput speeds of up to 1 million words per second. (A computer word will be defined shortly.) In addition, typical instruction speeds of most computers range from 100 to 600 thousand

[12] Some of the figures in this appendix were redrawn from Ref. A-1.

instructions per second (KIPS). The truly large multiprocessors and supercomputers range from 1000 KIPS o: million-instructions per second (MIPS) up to 10 or 20 MIPS. Since the mini is of prime importance in this paper, it should be noted that the mini here is defined as a computer whose cost is less than $50, 000, the word size is not greater than 18 bits, its core or dynamic memory is no greater than 32, 000 words, and its physical size is small, and in fact, will generally fit in a 19-inch equipment rack. In addition, it requires less than 30-A service at standard 117 Vac line voltage, and requires no special environment or air conditioning.

IV. DEFINITIONS AND JARGON

In general, minicomputers have the same basic elements found in their larger counterparts. The mini system elements fall into the following subsystems:

Processor (CPU). Most minis employ a single address structure, utilize two's complement negative numbers, and generally have no greater than 8 general purpose hardware registers. Usually, hardware multiply and divide is offered as an option. The minis usually have an interrupt handling capability (to be discussed) and a bus arrangement that allows transfer of data and control signals.

Memory. Most minis in the past were available with only core memory. Memory cycle times ranged from 600 nanoseconds to 1 microsecond. The new minis are now being made available with solid state dynamic memories whose cycle times range in the low hundreds of nanoseconds. The dynamic memories are faster than core, but more expensive at the present than core, and volatile in contrast to the nondestructive cores. In either case, however, memory blocks come in increments of 1024, 4096, or 8192 words. Most minis have a memory protect scheme that prevents the user from wiping out certain sacred areas of memory. In addition, a parity check is utilized in order to provide confidence of the integrity of the data retrieved from or deposited to memory.

Input/Output. Data is transferred to or from a peripheral device by one or more of three popular techniques: Processor Program controlled I/O through hardware registers in the CPU; or Direct Memory Access (DMA); or Direct Multiplexed Memory Access (DMC). The DMA technique of data transfer is the most efficient and fastest scheme in that it allows the CPU to function while data transfers occur.

Interrupt Structure. The interrupt structure concept is perhaps the most important parameter for minis used in real time control processes such as environmental test control. A computer interrupt is a means by which a signal occurs either internally to the computer or from a peripheral device that directs the computer to execute a special sequence of instructions as a function of the alerting signal. Priorities can be structured such that a signal from one device takes preference over a signal from a lower priority device. Figure A-4 is a flow diagram indicating the concepts of priority interrupts.

Software. Software for minis falls into four categories. The first category is program development software that is required by the user to develop his programs for particular applications. This includes editors, assemblers, debuggers, and utility routines and compilers such as BASIC or FORTRAN. Another category is Input/Output software for the system hardware and peripherals. This is the software that makes the terminals, line printers, reader, and punches work as well as the mass storage devices. A third category of software is the application programs. These programs are related to the task that the system is to perform and which is therefore unique to the particular system. The last category includes operating system software, also called the executive or system monitor, which tells the user what to do, when to do it, and what to do it with or to.

Minicomputer Peripherals. Peripherals for minicomputers can be separated into two broad categories - input devices and output devices. Most peripherals are used for both input and output functions. Input devices include keyboard terminals such as teletype (TTY) machines and Cathode Ray Tube (CRT) terminals, paper tape readers, fixed head disks, card readers and magnetic tape units including cassettes. Output devices include TTY's, CRT's, card and paper tape punches, disks, line printers, magnetic tape units and an X-Y recorder (plotter). For environmental test control, the following are the most popular peripherals:

(1) TTY or preferably a CRT terminal with hard copy capability.
(2) High-speed paper tape reader and punch.
(3) Low-cost disk (i. e. , the floppy disk) or cassette tape. It is highly desirable to have a line printer if your organization is going to assemble programs.

V. MINICOMPUTER SIZE, EXPANDABILITY AND FLEXIBILITY

Most minis fit in 19-inch equipment racks. Optional expansion chassis are available for system expansion. These expansion chassis include power supplies and cooling fans and can be used to house additional core or dynamic memory and additional logic required for controlling line printers, disks, and other added peripherals.

Many minis have a memory exchangeability feature that allows the faster dynamic semiconductor memories to replace the slower core memories by a board by board replacement (no mixing allowed). As faster memories become available, the old memory can simply be removed and replaced by a newer, faster one. With denser packaging technology, it is possible to replace a 4000-word circuit board by a single 8000-word board and later, no doubt, by a single 16,000-word board. Special interfacing can be housed in the expansion chassis.

VI. DEFINITIONS

The following definitions are listed here in an attempt to clarify some basic computer terms:

Bit. A bit is a binary digit, either 0 or 1.
Baud. This is a unit of rate of information. In general (but not always), a Baud rate can be divided by 10 to give an approximate character rate in units of characters per second (CPS). For example, 1600 Baud is about 160 CPS if the characters are ASCII characters.
ASCII. This stands for American Standard Code for Information Interchange. This is an attempt to standardize certain keyboard characters in terms of octal numbers.
Character. A character is a symbol that is defined under the ASCII standard. These characters are generated from keyboards, CRT terminals and printers. They include letters, numbers, punctuation, and special symbols including space, carriage return, line feed, etc.
Word. A computer word is a binary number which represents either data, an instruction, or an address. Minicomputer words are generally 8 bits, 12 bits, 16 bits, or 18 bits long. The bits can be grouped in three's and read as an octal number.
Byte. A byte is an 8-bit code. The code represents an alpha-numeric character or word. Generally two or more bytes represent a word, although by definition, 2 bytes can represent 2 words. Not all minicomputers are byte oriented.

VII. NUMBER SYSTEMS FOR COMPUTERS

Figure A-5 lists equivalent numbers in four number systems. Most mini programmers prefer to work in the octal numbering system since the binary bits can be grouped in three's and the octal number read directly from the grouping. Negative numbers are generated within the mini as two's complement. This is done by the machine by first one's complementing the bits (i.e., 0 becomes 1 and 1 becomes 0), then adding 1 to the results. An example is shown in Fig. A-6. The advantage of two's complement is simply that binary subtraction results in a sum process and, after all, adding is the thing a computer can do most efficiently.

VIII. THE MINICOMPUTER AS A SYSTEMS ELEMENT

The mini should be viewed by the systems designer as a small but nevertheless important element in the total system. The most common minicomputer systems are either man-machine or machine-machine oriented. The former consist of data acquisition, process control including environmental test control, and time sharing and problem solving systems, while the latter are either peripheral or remote terminal communications control systems. Figures A-7, A-8, A-9, and A-10 illustrate both systems. Note that the process control application requires a feedback configuration. This is a very important concept in environmental test control.

IX. INDUSTRY STANDARDS

Industry-wide standards for computers are either ill-defined or nonexistent. In most cases, the computer giant, IBM, has created its own standards, which to a great extent have, in turn, been adopted by the competition. These same IBM standards have also been adopted by the minicomputer industry. However, these transformed standards are generally for peripherals, not for computers. For example, transmission rate agreements are:

(1) Low speed:	110 to 300 bits per second.
(2) Medium speed:	300 to 2400 bits per second.
(3) High speed:	up to 50,000 bits per second.

Other standards (more properly, agreements) are 132-column line printers where speed is measured in lines per minute (LPM). Medium speeds are around 300 to 600 LPM, and high speeds are 1100 to 2000 LPM. The 80-column IBM Hollerith Punched Card is still one of the most widely used I/O media in the computer industry, although paper tape devices, being at the low end in terms of cost of I/O devices, are very popular in the minicomputer industry. Medium-speed paper tape punching is in the range of 50 to 75 characters per second, while the medium-speed paper tape reader range is from 100 to 300 characters per second.

X. MINICOMPUTER INDUSTRY

There are four main segments of the minicomputer industry - the main-frame manufacturers, peripheral manufacturers, software suppliers, and turn-key system suppliers.

A. Main-Frame Manufacturers

The minicomputer manufacturers were an outgrowth of companies supplying circuit boards to other manufacturers. In 1970, there were 40 to 50 companies manufacturing main frames Now, in early 1975, there are less than a dozen minicomputer main-frame manufacturers. A few of these companies are listed below:

Manufacturer	Computer
Honeywell	H 316, 516, 716
Hewlett-Packard	2100 A, 2114, 2115, 2116
Digital Equipment Corp.	DEC PDP 8, PDP 11
Interdata	Model 70

Manufacturer	Computer
Varian Data Machin ɜs	620, 73
Data General	NOVA

B. Peripheral Manufacturers

The second largest segment of the mini market is the ,eripheral manufacturers. Some of the mainframe manufacturers also make their own peripherals. The "peripheral" manufacturer buys or builds the end product and develops his own hardware/software controller. He then sells the entire system to the end user. His system may be an improvement on the comparable system sold by the main-frame manufacturer in that his peripheral control software may be more efficient, thus requiring less main-frame memory, or perhaps his system sells for substantially less. Peripherals include line printers, card readers, paper tape readers and punches, disk drives, plotters, CRT terminals and other keyboard terminals.

C. Software Suppliers

The third segment of the market consists of the software supplier who may be providing a more efficient, higher-level language or a general purpose modularized applications package that can be run on a widely used mini. Many such packages exist for the DEC PDP 8 and PDP 11 as well as for the Data General NOVA. Some of the larger companies supplying software to end users of minis are Computer Sciences, Planning Research, System Development Corporation, and Informatics.

D. Turn-Key System Suppliers

The fourth segment of the market comprises the independent turn-key system suppliers who compete head-on with the main-frame manufacturers. This supplier is generally at somewhat of a disadvantage since he must purchase a large part, if not all, of the hardware from the hardware manufacturer. The "value added" in this case is the applications software and the intimate knowledge of the applications area. Herein lie, at least to some extent, the suppliers of environmental test systems.

XI. MINICOMPUTER ARCHITECTURE

The architecture of a mini is the key to its performance and an indication of its relative utility and power. Machine architecture has a direct bearing on its operating characteristics, such as speed and efficiency, how instructions are carried out, and how easy or difficult it is to program the machine. The basic building blocks of all minis are its memory, the arithmetic and logic processor, the control logic, and the I/O logic. These subunits are interconnected by several lines that provide paths for data, control signals, and computer instructions. These lines are commonly called buses. Figure A-11 shows the bus line interconnections of the main building blocks of most minicomputers.

A. Memory

The memory unit provides the main storage for actual data as well as instructions that tell the control unit what to do with the data. Figure A-12 defines the memory organization concept. Perhaps the most difficult concept for a newcomer to computers to grasp is the difference between memory addresses (locations) and the contents of those addresses.

B. Arithmetic and Logic Processor

This subunit temporarily stores data received from memory and performs calculations and logic operations on this data. The arithmetic/logic processor contains one or more registers which, in turn, contain the data being operated on as defined by an instruction.

C. Control Unit

This logic controls the flow of data in the system, fetches instructions from memory, and decodes the instructions in one or more instruction registers. This unit executes instructions by enabling the appropriate electronic signal paths and controlling the proper sequence of operations performed by the arithmetic/logic and I/O units. Finally, it changes the state of the computer to that required by the next operation.

D. Input/Output Unit

This unit provides the interface and in some systems, buffering to peripherals connected to the computer, transferring data to and receiving data from the outside world.

It should be noted that some of the more recently introduced minicomputers treat memory as an external device and communicate with it over the same I/O bus, just as they would with an external peripheral device.

The central processing unit (CPU) is defined as the control unit, the arithmetic/logic processor, and all interrelated buses and registers. The basic elements of the CPU are:

(1) The Instruction Register(s).

(2) The Decoder.

(3) The Control Unit.

(4) The Program Counter (Pointer).

(5) The Adder and Comparer.

(6) The Accumula' r.

(7) The Status Register.

A typical minicomputer processing cycle can be divided into two subcycles - an instruction or fetch cycle and an execution cycle. A detailed description of each of these elements is beyond the scope of this paper (see Ref. A-1). However, the basic ideas of the central processing unit can be obtained by describing the chain of events that takes place during the fetch and execution cycles. During the fetch cycle,

(1) The control unit logic fetches next memory address n f.:om the program counter.

(2) The decoder decodes this address.

(3) The control unit logic fetches 'he contents A (an instruction) of memory address n.

(4) The control unit logic loads instruction A into the instruction register.

(5) The control unit logic increments the program counter by 1 for the next cycle, i.e., the counter acts as a memory pointer.

(6) The control unit fetches the contents of instruction register, A, and decodes the instruction.

The control unit is now ready to implement the second subcycle, the execution cycle. The steps followed in the execution cycle depend on the type of instruction to be executed. One of the most common instructions is addition or ADD. A typical ADD instruction is based on the following sequence of steps:

(7) A piece of data designated here as "a", stored in memory location m + n, is added to the contents of the accumulator.

(8) From the previous execution cycle, assume that the previously stored quantity in the accumulator is "b"; the new amount after additio. will be "a" + "b".

(9) The new contents of the accumulator are stored in location m + n, which will now contain "a" + "b".

The status register varies in size and complexity, but as a minimum, two single bit registers are used where the first stores the overflow from the accumulator, and the second register, called the test or link register, tests the condition of the accumulator.

Figure A-13 shows the architecture of a microprocessor. This is, for all practical purposes, a CPU on a chip. A 40-pin chip slightly over 2 inches in length and about 5/8 of an inch wide! This should give the reader insight into things to come in the computer industry in the near future.

REFERENCE

(A-1) Cay Weitzman, Minicomputer Systems, Structure, Imple-ientation and Application, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1974.

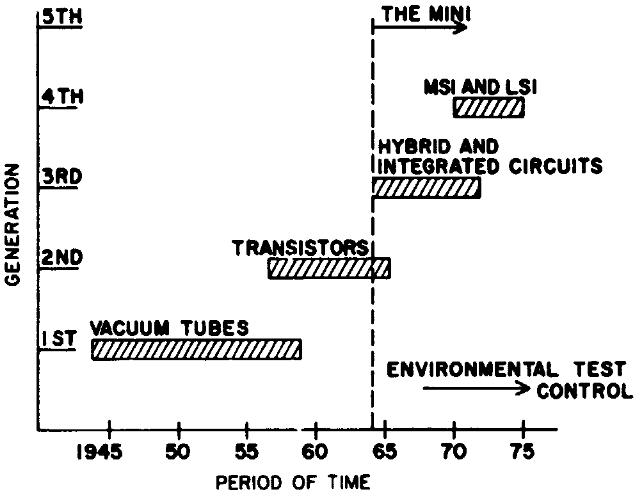


Fig. A-1. Evolution of computer technology

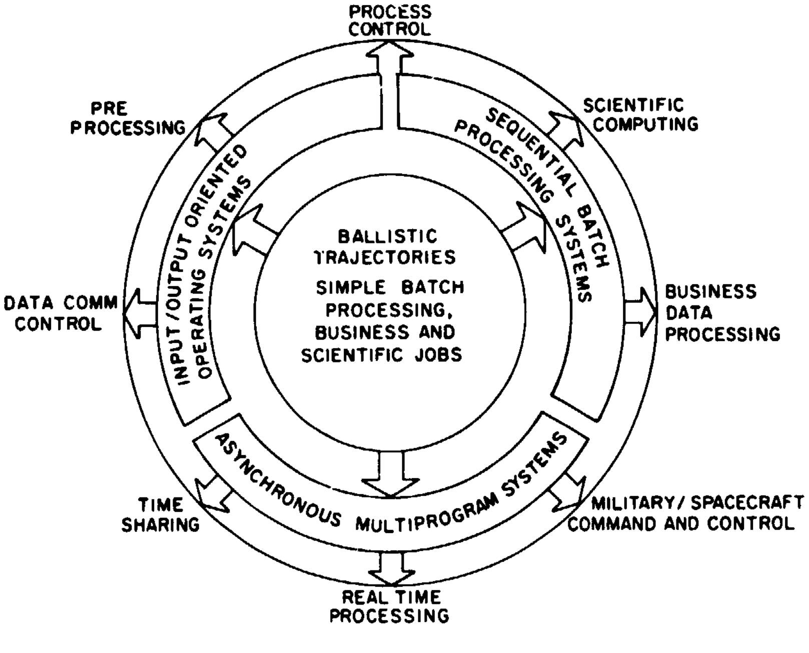


Fig. A-2. Evolution of computer applications

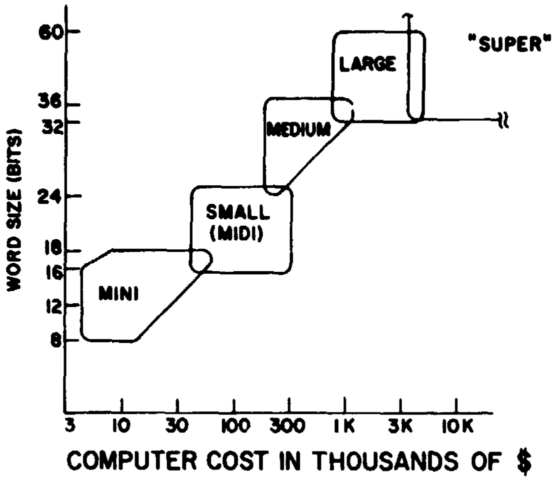


Fig. A-3. Classification of computers

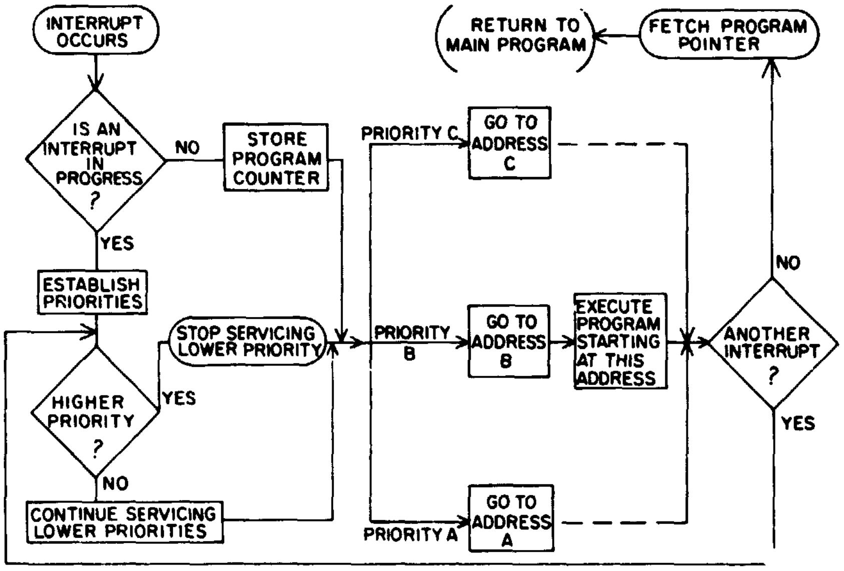


Fig. A-4. Interrupt concept

DECIMAL	BINARY	OCTAL	HEXADECIMAL
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Fig. A-5. Number systems used in computers

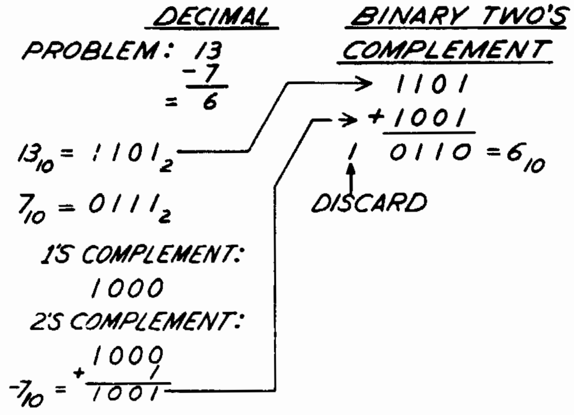


Fig. A-6. Two's complement concept

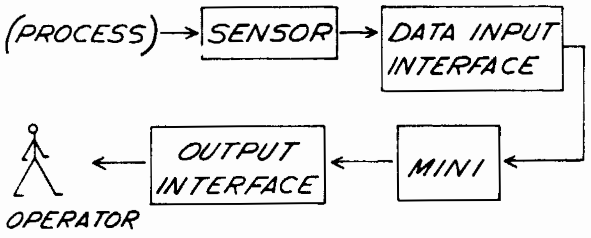


Fig. A-7. Man-machine system data adquisition

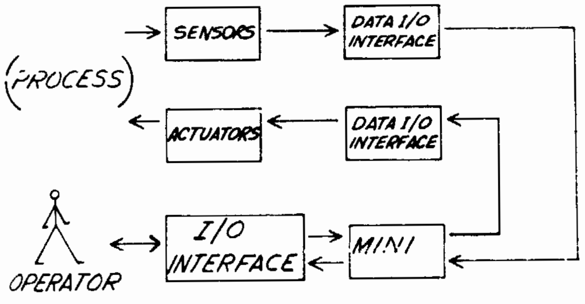


Fig. A-8. Man-machine system process contrcl

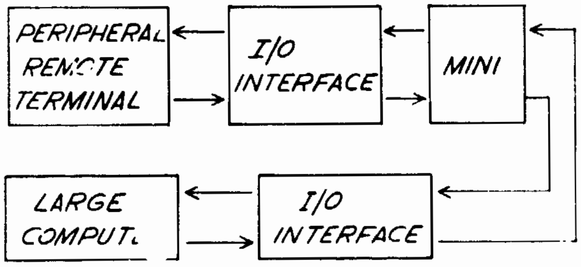


Fig. A-9. Machine-machine system front ends

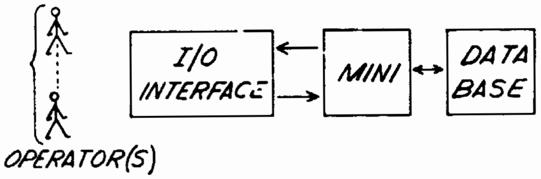


Fig. A-10. Man-machine system as interactive time-sharing, business data processing, problem solving

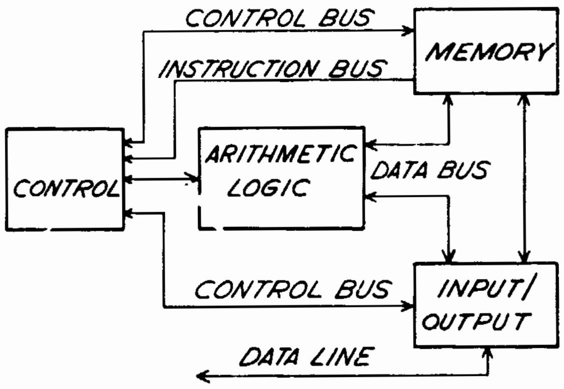


Fig. A-11. Minicomputer architecture organization

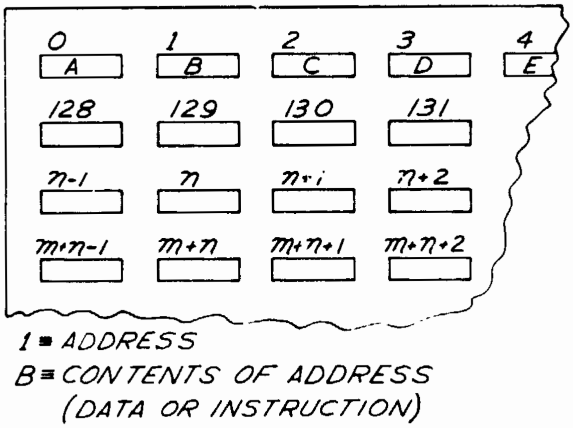


Fig. A-12. Memory organization

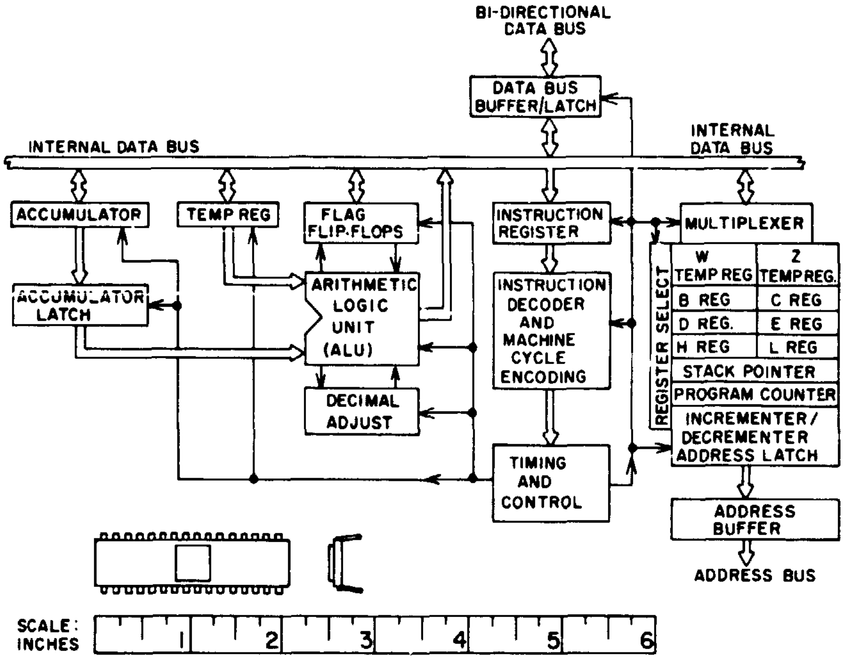


Fig. A-13. Microprocessor architecture

MINICOMPUTERS

Stephen A. Kallis, Jr.
Digital Equipment Corporation
Maynard, Massachusetts

A tutorial paper on minicomputers. Minicomputers share the characteristics of other general-purpose digital electronic computers in that they are programmable, can be used in batch, real-time, or timesharing mode. In addition, their degree of ruggedness, when compared to larger electronic computers, permits them to be employed in more severe environments than the larger machines; their cost is low enough to permit them to be employed as instruments in laboratory and test situations.

INTRODUCTION

Since its introduction in 1963 with the PDP-5, the minicomputer has had a significant impact in a variety of analytical fields as well as the more conventional "computer applications" such as statistical analyses, financial calculations, and the like. The minicomputers available today have analytical power comparable to the medium-scale computers of a decade ago, yet their costs are so low that a few people have purchased minicomputer systems for private use. Small businesses, and as important, small laboratories and research facilities are acquiring minicomputers to permit them to extend their activities into areas that it would have been impractical for them to have attempted only a few years ago.

A minicomputer is defined as a general-purpose electronic computer with a digital word length of between 8 and 18 bits, a minimum of 4,096 words in its memory, and a cost of less than $20,000 in its minimum configuration. For clarity, this definition will be refined further.

A "general-purpose" computer is one that can be employed for a variety of different applications. This is achieved through alteration in the programs stored within the computer's memory. Thus, the same computer could be used in connection with laboratory instruments at one time, for inventory control at another time, and for mathematical analyses at yet a third time, merely by reading new programs into the computer's memory for each new application.

This should be contrasted with a "special-purpose" computer, which can only be used for one sort of application. The flight computer aboard a spacecraft, for instance, is an example of a special-purpose computer.

The fundamental unit of information in a digital computer is a "bit," from Binary digIT. For operation in computers, bits are grouped into aggregations known as "words"; common minicomputer words are 12 and 16 bits long.

The capacity of a computer's memory is generally considered the number of words it can store. Common values for minicomputer capacity include 4,096, 8,192, and 16,384 words. All these numbers are powers of 2. In computer parlance, the term "K," when used for memory capacity makes use of the fact that 2^{10} is approximately equal to 1,000; therefore, the computer industry uses "K" to stand for 1,024 -- and by this definition, a computer with a memory capacity of 4,096 words would be said to have a "4K word" memory, and so forth.

Sometimes computer professionals work with another grouping of bits called "bytes." This term is a bit flexible, but usually refers to a group of bits less than one computer word in length, and frequently a submultiple of that word (e.g., a 12-bit word might be divided into two 6-bit bytes). To a number of specialists, though, "byte" seems to have a more fixed length, usually 8 bits. For discussion of the elements and applications of minicomputer systems, the term "byte" is not essential, and will be clearly defined in any areas where it is employed.

The cost factor of the definition of the minicomputer was strictly arbitrary and stemmed from the price of the original commercially available minicomputer, the PDP-5. The line of computers deriving from this machine, the PDP-8 family, has averaged a central processor cost reduction of approximately 10 percent per year, so the cost of the average minimum-configuration minicomputer system is approximately an order of magnitude below that of the defined minimum.

MODES OF OPERATION

There are two ways a computer can be used: As a "real-time" device or as a "non real-time" device. In computer terminology, "real-time" operations are those in which the computer is able to assimilate information concerning a process, process the data, and present the solution or solutions with sufficient speed so that the solution can affect the process under analysis. Although this definition implies a direct control feedback, intervention could take place through the agency of a human operator. The important concept for real-time operations is that the time spent in analysis should be extremely short when contrasted to the process being analyzed.

"Non real-time" refers to an operation that does not require immediate results. Traditional computer operations such as bookkeeping, financial analyses, generation of tables, and the like, are non real-time functions.

COMPUTER SYSTEM ELEMENTS

A computer system is composed of various physical devices, loosely categorized as "hardware," and computer programs, called "software." Within the category of hardware, there is the computer, peripheral devices, and interfaces. Within the category of software, there are application programs, general-purpose high- and low-level programs, and operating systems.

Treating the hardware first, there is the computer itself, generally called the central processing unit, or "CPU." Within this unit, all the electronics necessary to make the computer operate will be found. In many cases, the computer's memory is considered a part of the central processing unit, although in some schemes, the memory is treated as a separate element of a computer system (see Fig. 1).

In gross, the arithmetic and logical operations of the computer are performed by the central processing unit.

To extend the capabilities of the computer system, a number of devices, known as "peripheral devices," or "peripherals," are attached to the CPU. Peripherals fall into two main categories: "Input-output" (or I/O) and "storage" devices. The first category includes all units that establish a communication channel between the computer system and the user. Included in input-output devices are terminals, printers, plotters, card readers, paper tape readers, optical scanners, and paper tape punches. Because of the variety of available terminals, it is necessary to add that terminals may take the form of typewriters, supplying the user with a printed record of his communication with the computer system; or video terminals, where communications are displayed on a CRT screen. Those I/O devices producing tangible copies of computer communications are known as "hard copy" devices; those displaying communications are called "soft copy" devices.

Storage peripherals act to augment the computer's memory. Data or programs are stored on a recording medium and are available for callup by the computer in its operations. One of the most common storage media is magnetic tape. Magnetic tape is available on a number of different reel sizes, and is, relative to other media, inexpensive. Magnetic cassettes are considered as being in the "tape" category.

While magnetic tapes are inexpensive, they are also relatively slow in action. Another medium, the magnetic disk, operates at many times the speed of a tape: Where tape access to files may take seconds to achieve, access to those same files if stored on a disk can be achieved in 70 milliseconds or less (under 20 milliseconds for a fixed-head disk). This rapid rate of data transfer permits some varieties of disk to be used in timesharing applications, where the computer works with a number of terminals "simultaneously" -- actually by a form of time-division multiplexing.

A third magnetic medium used for data or program storage is the magnetic drum. These operate on the same principle as a magnetic disk, but with slightly faster access. Drums are not usually used with minicomputers.

Minicomputer peripherals, like minicomputers themselves, are both rugged and inexpensive. They tend to be usable in a less controlled environment than that required for large-scale computers. When the minicomputer was first developed, the only available peripherals were those designed for larger-scale computer systems, and the minicomputer system designer was frequently required to tie a peripheral to the central processor that cost four times as much as the CPU did, merely because that was the only unit available. Today, the low cost and durability of minicomputer peripherals are such that they are being used with large-scale computer systems.

A final type of hardware is the interface. When a computer is used to gather information from a laboratory instrument, or when one is used to control a process of some sort, appropriate electronics are

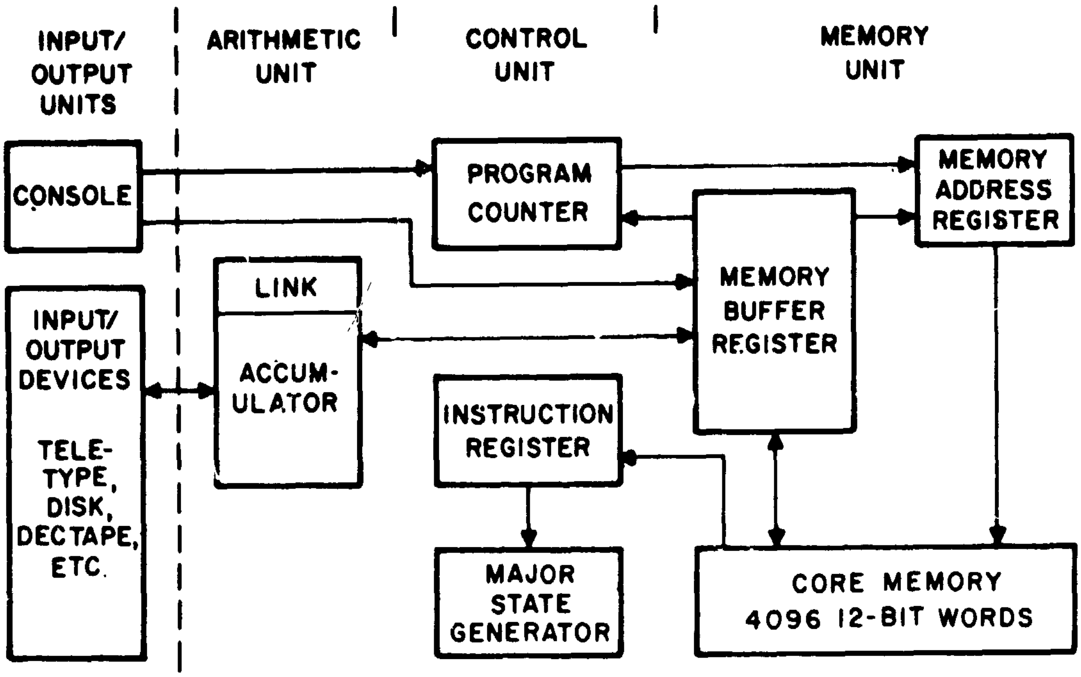
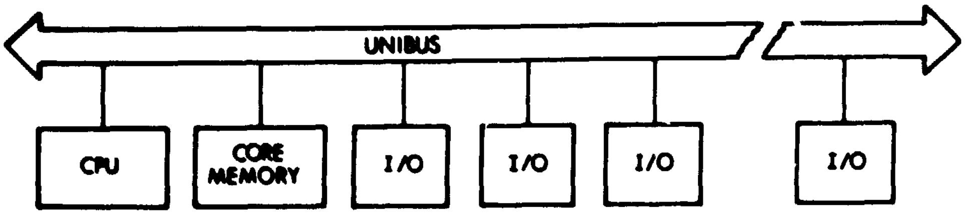


Figure 1.A - Minicomputer scheme in which the memory is treated as part of the central processing unit.



B - Minicomputer scheme in which the memory is considered separate from the central processing unit.

required to effect a connection between the computer and the process or instrument. In some cases, the interface can be a direct digital connection between the computer and a switch assembly (or equivalent). More frequently, however, some sort of conversion process is required to permit the computer to gather information from the process or generate instructions to it.

Many readings from instruments or sensors are in the form of electrical magnitudes (analog signals). To convert them into a form compatible with the circuitry of the minicomputer, an analog-to-digital converter is used. To convert information from the computer to analog signals required to control some process, a digital-to-analog converter is employed.

For long-distance connection between a terminal and a computer system, a special communication interface known as a "modem" (for MOdulator-DEModulator) is employed. A modem permits communication between a computer and a terminal (or between computers -- or terminals) over telephone circuits.

Programs, or software, fall into several categories. Some programs are very specialized, and permit a computer to perform a specific job only. These programs are called "application programs," because of their nature. There are many and varied application programs, ranging from simple tasks such as the generation of mailing lists to very complex programs such as those used to perform Fourier analyses on incoming waveforms. Application programs encompass a variety of disciplines, but each generally restricts a minicomputer to a single job or a single type of job.

A general-purpose high- or low-level programming language is used to develop application programs. A "high-level" programming language is one like FORTRAN, BASIC, COBOL, or FOCAL®. These languages are similar to English, and are relatively easy to employ. Most such languages are "machine independent, " meaning that the program written in, say, FORTRAN IV for one machine will run on another machine that was built by a different manufacturer. A high-level language is effectively independent of a minicomputer's internal organization, as far as the user is concerned, as is not the case with a low-level language.

A low-level language is written to reflect the actual organization of the minicomputer and its working parts. Such an "assembly language, " requires knowledge of the way a particular computer operates and is totally dependent upon the computer that it is being run on. Whereas a high-level language is easy to write and requires only a few instructions to execute a program, a low-level language requires utter precision in writing every step of action required in the program.

The advantage of a low-level language over a high-level language is quite simple: Although the instruction list is much longer for a low-level language the actual memory that contains the final program is much smaller than for a high-level language. The reason for this is that the computer only operates in binary instructions, and both the low-level language and the high-level language have to be transformed into binary instructions if the computer is to use them. In the case of the low-level language, this is a simple process; in the case of the high-level language, a large and complex program to translate the English-like program into binary instructions must also be present in memory. This large program (called either an "interpreter" or a "compiler, " depending upon its operation), because of its general purpose nature, occupies a large amount of a minicomputer's memory. Where memory is at a premium, the user may be forced to consider using a low-level language in order to conserve memory space.

Operating systems (also known as "executives") are programs designed to direct the operation of other program elements in a computer system. An operating system may be used to initiate a data transfer between peripherals, call up new programs (or parts of programs) for execution, or for library functions (keeping track of data files, for example). Operating systems have been available for computers for several years, and currently are quite sophisticated.

MINICOMPUTER USES

As noted before, minicomputers can be employed in a variety of different applications. In general, these fall into the two broad categories of data analysis and control. Administrative duties and data gathering without analysis can be associated with the former category; data communication and message switching, the latter. But there are two other fundamental ways a computer can be used: As a "stand-alone" unit, or as a part of some sort of computer hierarchy.

The term "stand-alone" implies the computer use: The computer is used as an independent unit, tied into no other computer system. The relatively advanced software available today makes such an approach practical. Today, minicomputer systems can run ANSI-compatible FORTRAN IV programs in as little as 8K words of memory, with the assistance of the appropriate peripheral(s) such as a mass storage device. This degree of power means that computer systems that can fit atop a desk can deliver power to the engineer that only a few years ago was available only through use of a large-scale centrally located computer system.

Stand-alone computer systems permit a degree of protection for a facility's operations that is difficult to achieve by other means. A number of stand-alone systems scattered throughout the facility result in a distribution of the total computer power among a number of independent stations. The workload of one system will not necessarily interfere with the operation of another system, and in the event of a system failure, only the one station will be affected, not every station.

Where a stand-alone computer is used for only a fraction of a workday on a single application, it is frequently possible to re-program it for other applications in its idle time. Some facilities have mounted their smaller minicomputer systems on casters so that they can be wheeled between laboratories, permitting them to be shared during the course of a workday.

Minicomputers can also be used in hierarchies. A number of minicomputers can be connected, via communication lines of some sort, to a larger computer. In this sort of arrangement, the minicomputer acts in a semi-independent mode, transmitting data to the larger computer but handling ordinary problems itself. If, however, it encounters an extraordinary problem, beyond its capacity to solve -- or beyond its capacity to solve in an acceptable period of time -- it may format its problem in a "predigested" form and forward the problem to the larger computer to solve. The solution is returned to the minicomputer, which then handles the solution as if it had performed the necessary operations.

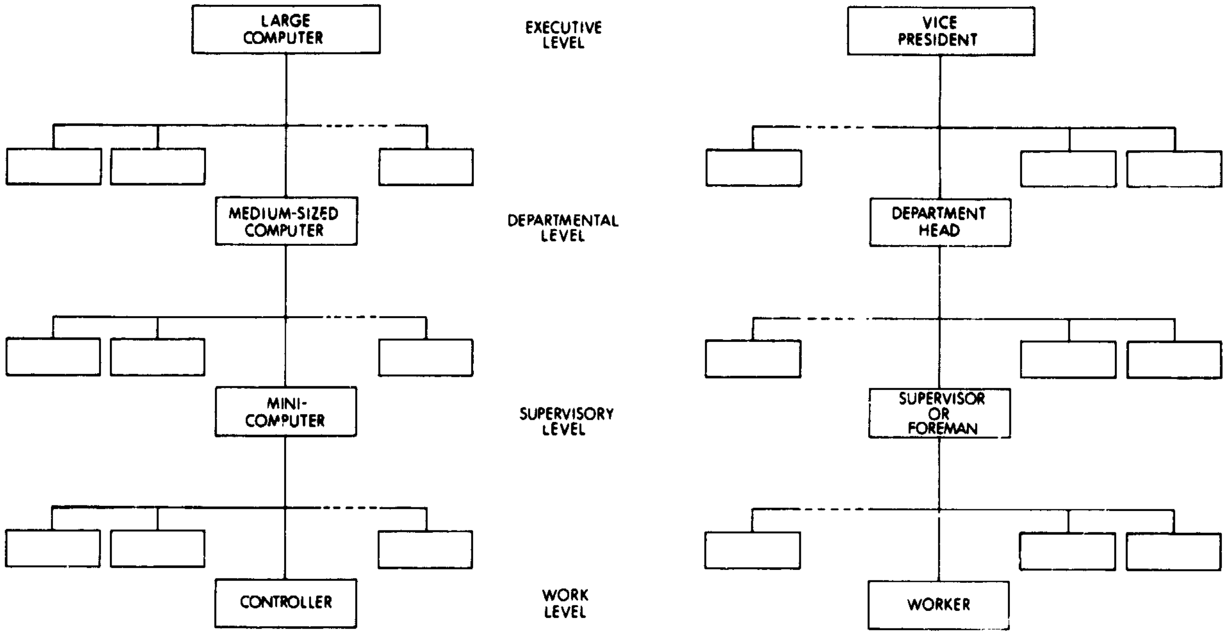


Figure 2. - A computer hierarchy. In this example, the computer hierarchy resembles the corporate structure of a manufacturing facility, with each level of computer activity mirroring an equivalent human activity.

The larger computer may in turn be tied to an even larger computer. The minicomputer can, in turn, be linked to even simpler automatic devices, such as programmable controllers. Such a hierarchy parallels the hierarchy found in large organizations. (See Figure 2 above)

While computer hierarchies are interlinked, there is also a certain degree of independence between the various levels. Minicomputers, for example, ordinarily will work independently of the medium-scale computer. The only times that communication normally will occur between these machines is either when the medium-scale computer interrogates the minicomputer for information, or the minicomputer forwards data to the larger machine to obtain the solution of a problem. The larger machine can forward information to the executive machine; it also can coordinate the operation of the minicomputers it is monitoring. The effect is the most optimum use of computers in a facility, with each being used at its maximum efficiency, with minimum interference from each other. Even with a short breakdown in communication, the elements of the hierarchy would be able to operate independently.

CONCLUSION

The minicomputer has evolved into a powerful, rugged, and reliable unit that can be used as a supercalculator, an analytical device that can be attached directly to a process, a control device, an administrative device, or an element of a computer hierarchy. Historically, the cost of minicomputers has diminished while their power has been maintained, or has actually increased. Software for minicomputers has become highly sophisticated, and minicomputers can perform operations today that only a few years ago were the exclusive province of much larger computers. The result is a tool that can be incorporated into a wide variety of systems for a large number of varied applications.

-end-

Note: "PDP" and "FOCAL" are registered trademarks of Digital Equipment Corporation.

DIGITAL ANALYSIS AND CONTROL IN THE VIBRATION LABORATORY

Peter Moseley
Hewlett-Packard
Digital Signal Analysis Group
Santa Clara, California

A discussion of the differences between the traditional analog instrument and the modern mini-computer based digital instrument is given. Examples of the kinds of measurements and new capabilities provided by a digital instrument are presented. Included is a brief discussion of how vibration control is implemented in a digital instrument/system. Finally, a discussion is presented on the impact of digital instruments on the typical vibration test laboratory.

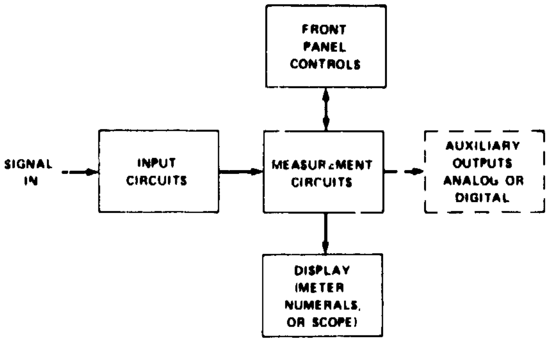
INTRODUCTION

The advent of the mini computer has allowed the basic concept of an instrument to change, and this is why digital-based instrument/systems are having such a dramatic effect on the vibration test laboratory.

This paper summarizes the important difference between analog and digital instruments and discusses some of the advantages of the digital approach. The use of memory to save data and numerical computation to make the measurement allows a single instrument/system to perform a wide variety of tasks in the vibration laboratory. These capabilities range from PSD analysis to transfer functions (and therefore modal analysis) to complete random, sine and transient control.

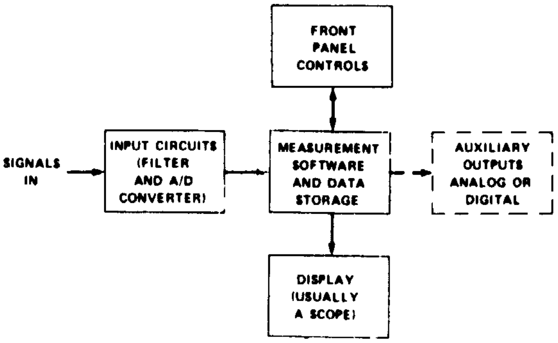
Analog and Digital Instruments

Figure 1 is a block diagram of a typical analog instrument.


FRONT PANEL CONTROLS
SIGNAL IN
INPUT CIRCUITS
MEASUREMENT CIRCUITS
AUXILIARY OUTPUTS ANALOG OR DIGITAL
DISPLAY (METER NUMERALS, OR SCOPE)

The instrument includes input circuitry, front panel controls and some form of display; for example, a meter movement, numerals or a cathode ray tube (CRT). The measurement is actually performed using fixed purpose analog circuitry which may typically be made to operate over several ranges, but still performs only one fundamental type of measurement. More sophisticated analog instruments provide "plug-ins" to extend the capability, and offer various outputs to transfer the measured information to another device; e.g., a recorder, printer or a computer.

Figure 2 is a block diagram of a typical digital instrument.


FRONT PANEL CONTROLS
SIGNALS IN
INPUT CIRCUITS (FILTER AND A/D CONVERTER)
MEASUREMENT SOFTWARE AND DATA STORAGE
AUXILIARY OUTPUTS ANALOG OR DIGITAL
DISPLAY (USUALLY A SCOPE)

Superficially, it looks much like an analog instrument, but there are many important differences both for the manufacturer and the user. The input circuitry consists of anti-aliasing filters and analog-to-digital converters. The front panel controls usually are no longer knobs, but more generally are push-buttons. These can range from alphanumeric keys on a CRT terminal to pushbuttons on a control panel (see Figure 3).

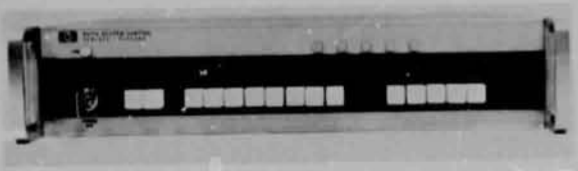


FIGURE 3

A Digital Instrument/System Control

The display can range from a CRT to a full graphics capability terminal. The auxiliary outputs can include complete graphics capability.

The fundamental difference between analog and digital instruments is in the way a measurement is made. Provided the anit-aliasing filters are properly set, the data that is gathered by the A/D converter and stored in the instrument memory represents <u>all</u> the information available about that signal during the time interval it was observed.

Once the data has been sampled, the software in the instrument determines the form in which the information is presented to the user, i.e., the "measurement." The power of the digital instrument lies in the fact that once the data has been properly sampled virtually any measurement may be made.

The information that is available is determined by the sampling parameters used to set the ADC which in turn is governed by the relationships shown in Figure 4. These relationships are laws of nature and do not vary between instrument manufacturers nor between vibration laboratories.

The speed of the analog-to-digital converter determines the bandwidth of the instrument, and the number of bits (and the computer word size) determine the dynamic range. For example, with a 10-bit ADC and a 16-bit word size a narrow band power spectrum may be computed with an 80 dB dynamic range.

Table I lists a representative set of measurements which may be made from the keyboard of a digital instrument. All of these represent the same information transformed by one process or another to the most useful form.

<u>Memory</u>

Another important difference between analog and digital instruments is the use of digital memory. This allows the original data to be saved and the "measurement" to be performed after the time waveform itself is gone. Further, many "measurements" may be made on the same data. For example, it is possible to compute, among other things, the histogram, power spectrum, auto correlation function, and rms power of a single set of random data. To make a similar number of measurements using analog instruments would require many instruments making measurements either simultaneously or one after the other on recorded data.

<u>Speed, Bandwidth, Overlapped Processing and "Real Time"</u>

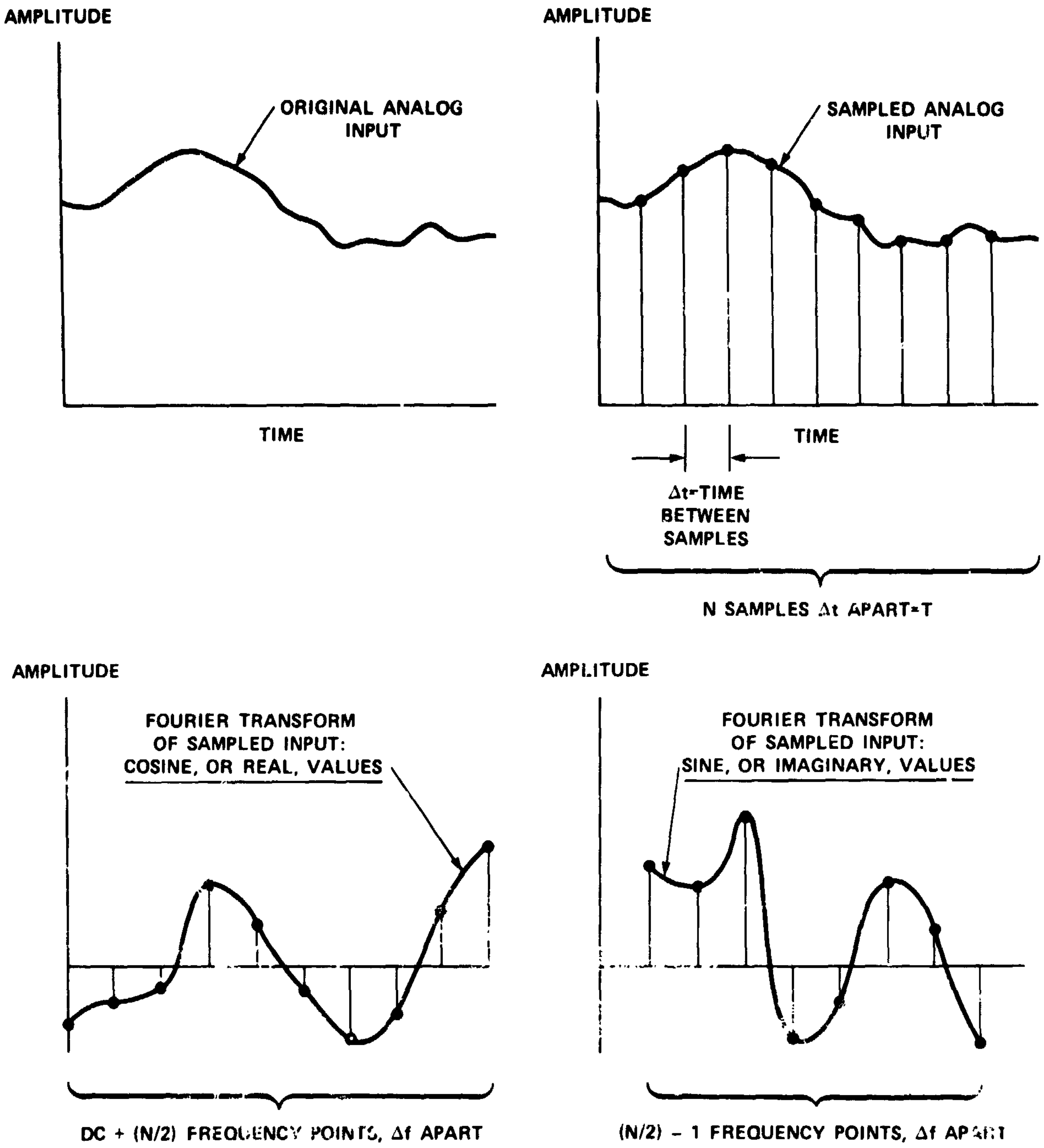
Ocassionally, it is desirable to make the "measurement" (i.e., sample the data and perform the computations necessary to transform the data to the desired form and display it) in "real time"; that is, without loss of data over many contiguous observation times T. In this case, the performance of the machine is limited by the amount of information (bandwidth) and processing speed of the instrument. Processing speed depends upon the computer cycle

TABLE I

TYPICAL MEASUREMENT CAPABILITY
OF A DIGITAL INSTRUMENT/SYSTEM

RMS Power
Peak
Average
Linear Spectrum
Auto Power Spectrum
Auto Correlation
Histogram
Integration
Differentiation
Band Selectable Fourier Analysis
1/N Octave Analysis
Shock Spectrum Analysis
Amplitude Modulation

Frequency Modulation
Phase Modulation
Hilbert Transforms
Cross Power Spectrum
Cross Correlation
Transfer Function
(Transmissibility, frequency response, impedance)
Impulse Response
Convolution
Coherence (signal-to-noise)
Multiple Coherence



BASIC SAMPLING LAWS

$T = N\Delta t$
$T = \frac{N}{F_s}$
The observation time increases as the amount of data increases, and decreases as the maximum frequency increases

$\Delta f = 1/T$ The longer the observation, the finer the frequency resolution. (ΔF is the lowest observable frequency.)

$F_{max} = \frac{1}{2\Delta t}$ The maximum observable frequency is 1/2 the sampling frequency.

FIGURE 4

time and upon the organization of the software. In machines equipped with microcoding capability up to a 10 to 1 speed improvement may be obtained over conventional machine language code by implementing certain highly repetitive operations in microcode(1).

When making measurements on random vibration data it is desirable to get the maximum variance reduction per unit of time. The greatest variance reduction comes when the data frames are overlapped prior to performing the Fourier Transform. Overlapping capability is particularly useful in analyzing low-frequency random data, i.e., less than 500 Hz, because the record time T becomes relatively long and to get the desired variance reduction a great many averages may be required. By overlapping, the time to get the equivalent variance reduction is significantly reduced. For example, if the measurement bandwidth is 50 Hz and 100 averages are required, overlapping can provide a 30 to 1 reduction in measurement time.

As the bandwidth is increased, the processing time becomes a larger and larger fraction of the data input time T. When they are just equal, this is the "real time" limit of the machine for the given measurement. Beyond this limit data is lost. For example, the real time limit for a typical Fourier Analyzer is 3000 Hz when performing a power spectrum average.

Mass Memory

If the process under observation can be considered stationary, then the loss of data above the real time limit is of little consequence unless there was a limited amount of it to begin with and not enough averages can be made to achieve the desired variance reduction. In this case, some alternative procedure is desirable to capture all of the available data. The data must be put into a mass memory large enough to hold the data. This can be done if the digital instrument has the capability to digitize and send the data to a magnetic disc or tape in "real time" that is without loss of data. This capability is referred to as "ADC throughput." Once the data is on the disc or tape, then the measurement can be made after the fact using all of the data.

The bandwidth limitation is determined by the throughput rate to the disc or magnetic tape. A typical specification for this is 39 KHz throughput rate per channel, which means that single-channel measurements can be made over a bandwidth of 19 KHz, and dual-channel over a 9 KHz bandwidth.

A digital instrument equipped with a magnetic disc or tape not only can have throughput capability but also general data and program storage capability. This extends the usefulness of the machine and provides, among other things, the means for establishing a digital data base for such applications as modal analysis and signature analysis.

Most vibration laboratories record test data on analog magnetic tape and perform detailed analysis after the test is complete. Although digital magnetic discs and tapes are extremely useful tools in conjunction with a digital instrument, they do not replace the function of analog tape recordings of test data. These records are often just that—records—and they need to be kept in an inexpensive form. In addition, to digitally record, say, 200 channels of data from a four-minute spacecraft vibration test out to a bandwidth of 2.5 kHz is not, as yet, practical, as it would require throughput rates in excess of 2 megawords/second to record on a single device. If parallel devices are used, it is feasible, but expensive. In addition, if phase information is to be maintained between all the channels, then they must be simultaneously sampled. This would imply multiple multiplexers slaved together and, again, it becomes expensive.

For modal surveys, however, where the test levels are lower and the test duration can be lengthened, then the digital throughput method works extremely well as a means of gathering and maintaining the data in an organized and readily accessible form.

Some Considerations in Making Measurements with Digital-Sampled Data Instruments

As noted before, almost any measurement can be performed on the sampled data. The measurements, to be properly made, must rest on a strong theoretical background. Yet, the user of the instrument should not need to be intimately familiar with the theory, so long as he can properly make the measurements he needs.

Fixed Resolution Bandwidth Measurements

A practical problem in this regard comes up when measurements are made on random data. For example, it is an easy matter to measure the rms power of a signal using digital techniques and with an ordinary true rms meter. But what if they disagree? Which instrument is right? Most often they are both right. The difference is the bandwidth of the instrument. The true rms meter might have a bandwidth from 10 Hz to 1 MHz, and the digital system might be set to have a bandwidth from DC to 2 KHz. Clearly, the digital system will measure less rms power.

This same difficulty arises in comparing spectrum analyzer outputs. There are often differences between wave analyzers, "speed-up" analyzers and digital FFT-based instruments. The differences almost always come back to a failure on the part of the person making the measurement to make the measurements using comparable resolution bandwidths and properly normalizing the results.

Figures 5 and 6 show two measurements of a sine wave with harmonics superimposed on broadband random noise. The measurement in Figure 5 was made using a spectrum analyzer which uses analog sweep techniques. Figure 6 shows the same measurement made using a Fourier Analyzer which uses FFT techniques. The measurements were not made to quite the same overall bandwidth nor to the same resolution bandwidth.

Figure 7 was made with the analog analyzer using a 3 Hz bandwidth and now we see how closely the two agree.

The important point is that the different filter bandwidths are performing different convolutions on the data and that it is the area under the curve that gives us the information, not the peak. Notice that the peaks in Figures 5 and 6 differ considerably, but the areas are approximately the same.

Variable Resolution Bandwidth Measurements

Many specifications such as MIL 810C allow random data to be measured using two different bandwidths, say, 25 Hz from 20 to 2000 Hz and 50 Hz from 200 to 1000 Hz.

What do you do if your digital instrument can only give you constant bandwidth measurements? In general, there is no difficulty getting enough resolution, but sometimes more than enough is considered too much. A very narrow line containing only a small amount of energy may peak up beyond the test limits when analyzed with more resolution than specified.

There are several solutions to this problem. One is to measure the data twice using two different bandwidths and merge the data together to form a composite spectrum. Another is to measure the data with a single measurement of narrower bandwidth than required and then apply a smoothing function to the lower frequency portion to effectively broaden the filter shape. Other techniques are possible using digital filtering methods.

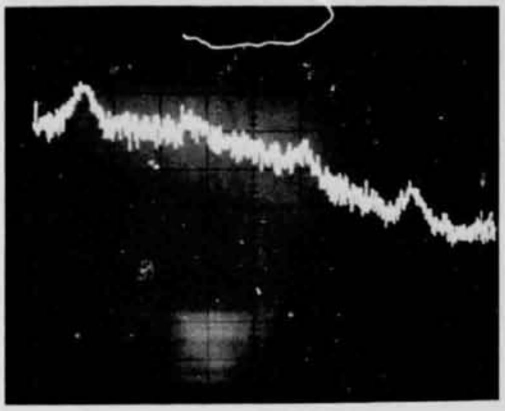


FIGURE 5

Analog Spectrum Analyzer
Resolution Bandwidth = 30 Hz

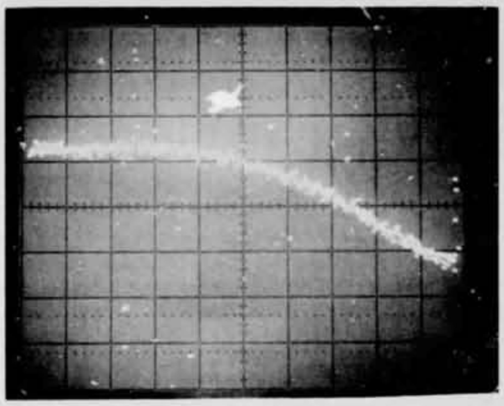


FIGURE 6

Digital Spectrum Analyzer
Resolution Bandwidth = 1.02 Hz

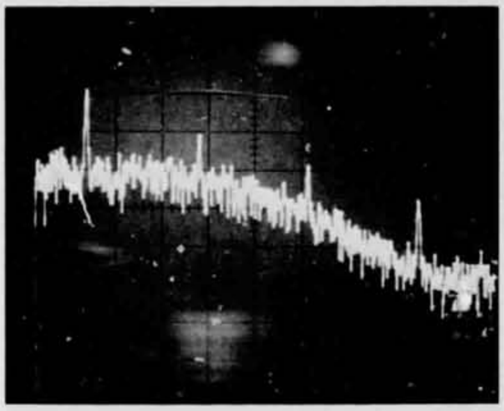


FIGURE 7

Analog Spectrum Analyzer
Resolution Bandwidth = 3 Hz

This example emphasizes the unique capabilities of a digital instrument to "massage" data to suit the needs of the user. However, this capability must not be incorrectly or improperly used or erroneous results may be obtained.

SOME NEW CAPABILITIES

Transfer Functions

The concept of a transfer function is old, but its use in the vibration test laboratory is still in its infancy. Partly, this stems from the fact that the traditional methods for making transfer function measurements are quite slow. A transfer function computed using baseband Fourier analysis techniques (that is, measurements which cover the range from DC to some maximum frequency) can be made in a matter of seconds and can provide a great deal of useful information.

An example of this is the use of low-level random noise excitation to measure the transmissibility between critical parts of a spacecraft before and after sinusoidal qualification testing. A comparison of the data before and after quickly reveals if the structure has changed and at what frequency. This technique has pinpointed trouble that might possibly have escaped detection on several occasions.

The reason the technique is fast is that a broadband stimulus is used to make the measurement and all the frequencies are measured at the same time. The stimulus can be random or pseudo-random or even a transient(2). Small hammers with load cells mounted on them make ideal "impulse generators" and are now available commercially.

The use of transients for making transfer functions has several advantages. First, it is easy to set up and, second, the measurements can be made very rapidly. Complex shaker fixturing is not required, and it provides a very good "quick-look" capability. A disadvantage comes when the structure under test is very non-linear or requires a carefully controlled force input. Often, the problems with non-linearities may be overcome by preloading the structure. More often than not, controlled force is not required because of the large dynamic range capabilities of the digital instrument.

Another example of the utility of easily made transfer functions is fixture verification. In general, fixtures are linear and pseudo-random noise can be efficiently used to excite the fixture and measure its transmissibility and verify the design.

Modal Analysis

The most general use of transfer function measurements comes in modal analysis, which is discussed in detail in References (3), (4) and (5). There it is pointed out that the transfer function contains all of the information necessary to identify the modal parameters of a system; that is, the resonant frequency, damping, and complex residue. Further, a matrix of transfer functions can be used to completely model a mechanical system. By examining the complex residues of a series of transfer functions, the mode shape vectors can be obtained and displayed in animated form.

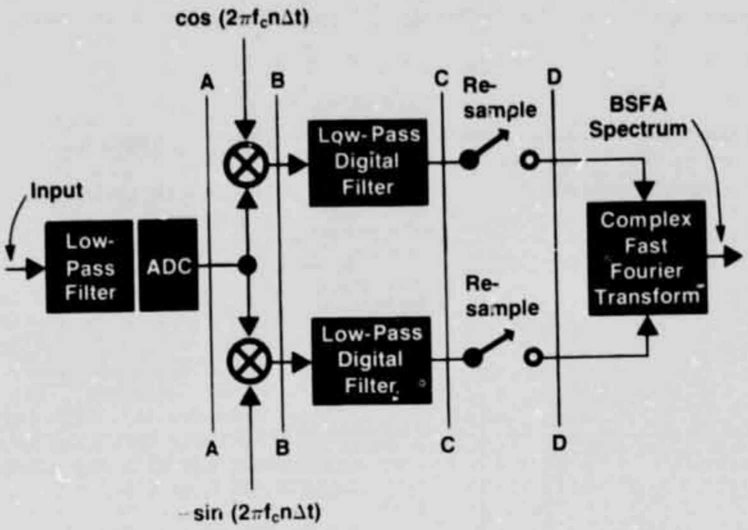
Resolution

One of the objections raised about digital Fourier transform-based transfer functions is resolution. The traditional transform is a baseband measurement; that is, it provides uniformly spaced frequency samples from DC to some maximum frequency. What do you do if you need resolution of a millihertz at a kilohertz? Baseband digital methods would require an enormous data record and would take a long time to process a lot of unwanted information. A new solution is Band Selectable Fourier Analysis.

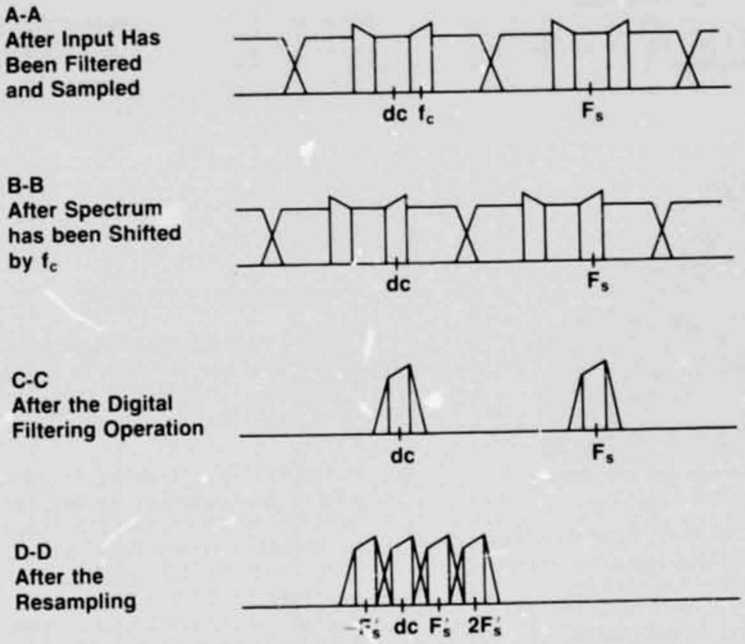
Band Selectable Fourier Analysis

Band Selectable Fourier Analysis, or BSFA, is a measurement technique that makes it possible to perform Fourier analysis over a frequency band whose upper and lower frequencies are selectable. BSFA provides improvement in frequency resolution greater than 100 times compared to standard Fourier analysis without increasing the number of spectral lines stored in the computer. The dynamic range is maintained at 80 dB when the major signal is within the selected band and is increased to 90 dB or more when the major signal is outside the band. The dynamic range is considerably greater than that obtained with analog hetrodyne techniques because of the sharp filters that can be implemented digitally(6).

The way BSFA is achieved is to multiply the sampled data by a sine and cosine wave at the selected band's center frequency and then digitally filter the data, resample it and Fourier transform the result (see Figure 8). The sine, cosine multiply produces a complex time waveform shifted so that the center frequency is now at DC; the digital filtering removes the unwanted information outside the selected bands; and resampling, from the filtered data stream, causes the maximum frequency of the new data to be one-half the bandwidth of the selected band. The Fourier transform (now complex) then coverts the information to the frequency domain. Once in this domain, power spectral density and transfer functions can be computed. It should be pointed out that the sampling laws of nature described above still



Consider what is happening in the frequency domain after each of the operations shown above. The region of interest is highlighted by giving it a ⎍ shape. The remainder of the spectrum is assumed to be flat.



This signal is then Fourier transformed into N equally spaced spectral lines between $-F_s'/2$ and $F_s'/2$.

Important BSFA Relations

$$\Delta f' = F_s'/N = F_s/n \cdot N = 1/T'$$

Where:

F_s' = Resampling Frequency
N = Number of Frequency Lines
F_s = Original Sampling Frequency
n = Integer Determined by F_s and BSFA Bandwidth
T' = Length of Time Record

FIGURE 8. BSFA signal flow. A frequency shift by f_c centers the analysis band on the frequency of interest and digital filtering removes information outside the band of interest.

apply and that to achieve fine resolution still requires a long measurement time. For example, to obtain 1 millihertz resolution requires 1000 seconds of data. Figures 9 and 10 show the advantages of BSFA in power spectrum measurements. Figure 11 shows a twc-channel transfer function BSFA measurement.

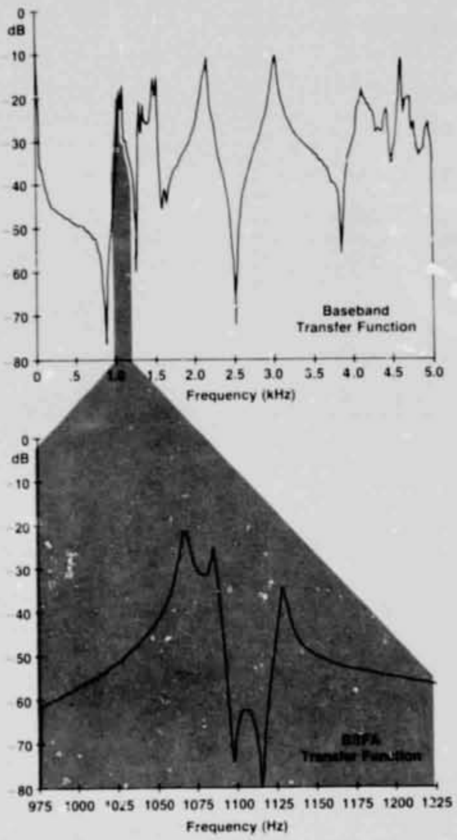


FIGURE 11. In this mechanical impedance measurement on a squeaky automobile disc brake rotor, BSFA made possible an accurate determination of the modes of vibration and the damping associated with each mode.

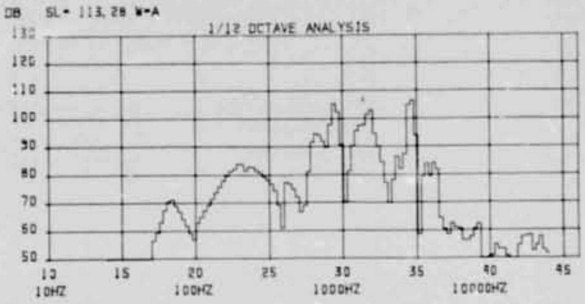
With BSFA capability, it is now an easy matter to make a quick baseband measurement, decide where to look more closely and then zoom in on it. (Hence, the name Zoom transform is sometimes used to describe BSFA.) No longer is resolution a deterent to the use of the transfer function in the vibration test laboratory. Indeed, the BSFA technique is essential to successful modal analysis using broadband transfer function techniques.

The BSFA measurement can be made on data saved on a mass memory using ADC throughput capability. This means that one set of data can re-examined with any resolution up to that dictated by the record length itself.

1/Nth Octave Analysis

Just as band selectable Fourier analysis makes use of digital filtering techniques, so does 1/N octave analysis. 1/N octave analysis is a measurement technique that utilizes convolutional digital filters to separate the spectral components of a sampled waveform into 1/N octave spaced bands of 1, 1/2, 1/3, 1/6 and 1/12 octave spacing. Because of the digital nature of the instrument, the information is stored in memory in 1/12th octave and the final result may be viewed in any of the octave spacings with any standard weighting; i.e., A, B, C or D.

Figure 12 shows a plot of a typical 1/12th octave measurement performed on acoustic data.



Shock Response Spectrum

Another measurement that can be implemented using digital filters is the shock response spectrum. Unlike the Fourier transform, this is a non-linear, irreversible process.

Historically, it predates even most electronic analog instruments, having come from Biot's efforts to determine the frequency spectrum of earthquakes using tuned mechanical "filters." This "poor-man's" spectrum analyzer has been emulated in both electronic analog and digital instruments, but seems archaic now that so many more powerful tools are available (7). However, its use persists and the fact that it can be implemented in a digital instrument demonstrates the versatility of such devices.

Special-Purpose Measurements

As should be clear by now, the measurement capability of a digital instrument system lies in the software that implements the measurement. If a system has mass storage capability such as a disc then a great deal of software can readily be loaded into the computer to change the system capabilities. Special-purpose measurement packages such as signature analysis and modal analysis can be implemented this way. These make use of the basic measure-

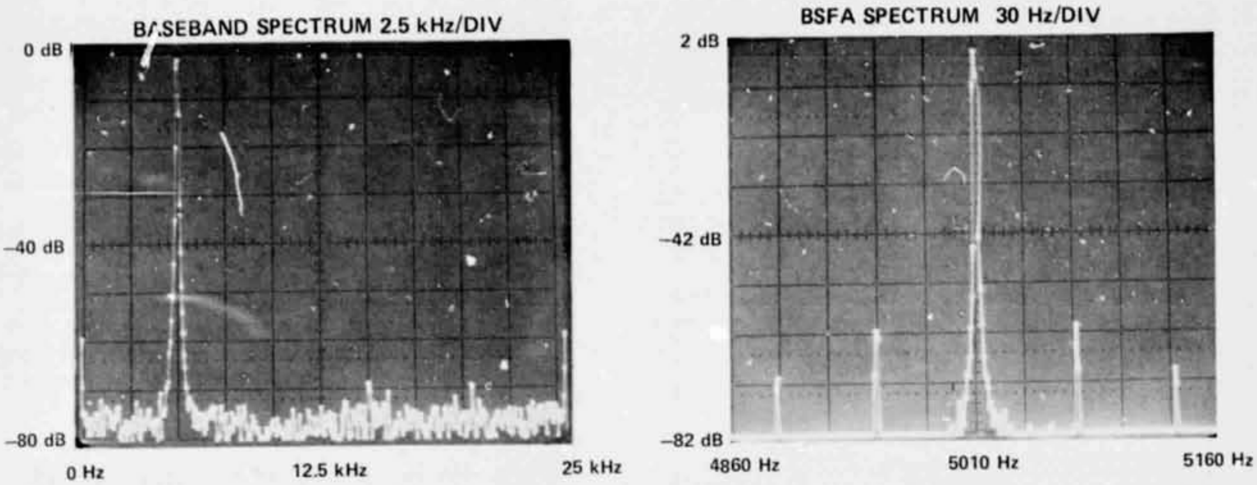


FIGURE 9. The resolution and dynamic range improvements of Band Selectable Fourier Analysis (BSFA) are demonstrated by these power spectra of the output of a sine wave generator. The conventional baseband spectrum shows the 5-kHz fundamental and the third harmonic at -69 dB. The noise floor is 72 dB down. The BSFA spectrum shows that the fundamental is actually at 5010 Hz and that there are 60-Hz and 120-Hz sidebands. The noise floor is greater than 80 dB down.

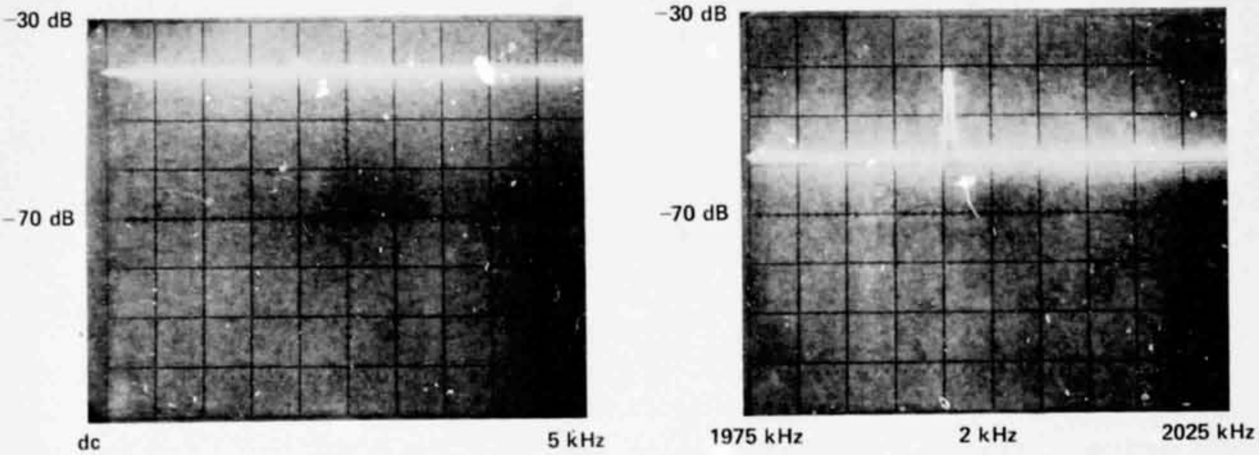


FIGURE 10. In the baseband spectrum of a sine wave buried in white noise the sine wave appears only 2 dB above the noise after 10,000 averages. In the BSFA spectrum after only 100 averages, the sine wave is 17 dB above the noise and consequently is much more detectable. BSFA's 100 x improvement in resolution results in a reduced mean value for the white noise; however, the noise variance is greater because fewer measurements were averaged. The two measurements required the same total time.

ment capabilities described above, but present the capabilities to the user at a much higher level. Thus, a single button push can produce an RPM spectral map or an animated mode shape.

Vibration Control

As if all of the above measurement capabilities aren't enough, a digital instrument/system can provide vibration control capability including sine, random and transient (shock) control. Figure 13 shows how a digital analysis instrument is turned into a digital vibration control system.

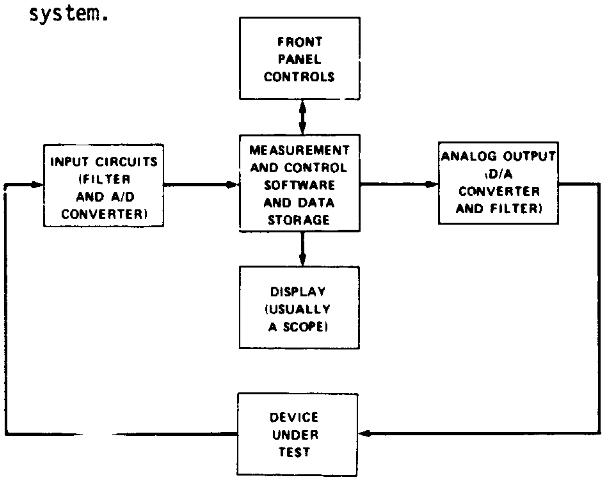

FRONT PANEL CONTROLS
INPUT CIRCUITS (FILTER AND A/D CONVERTER)
MEASUREMENT AND CONTROL SOFTWARE AND DATA STORAGE
ANALOG OUTPUT (D/A CONVERTER AND FILTER)
DISPLAY (USUALLY A SCOPE)
DEVICE UNDER TEST

FIGURE 13

Block Diagram of
Digital Vibration Control

An nalog-to-digital converter and a filter, along with special software, are added to give signal generation capability to the system.

Although the basic elements are the same as those available in the general purpose instrument/system, the uses and requirements of a vibration control system are different and more emphasis needs to be given to simple operator controls to avoid mistakes and clarify the operation. The system control keyboard is used for this purpose (see Figure 3).

If the vibration control system is disc-based, changing from Fourier analysis to random to sine or to transient control can be almost instantaneous.

The successful implementation of digital signal generation and control over 5 KHz bandwidth requires a high-performance computer. The overhead required to generate the signals takes away from the processing time available for measurement and control and, if it becomes too great, makes for a poor control situation.

In the Hewlett-Packard Vibration Control System, the generation of the random signal(8) and the sinusoidal signal are both implemented making use of the microcode capability of the machine.

The use of microcode to keep the signal generation overhead to a low level, as well as perform several other time critical operations, makes it possible to implement 4-channel random control and up to 12-channel sinusoidal control. Four-channel random control is performed by averaging four power spectra computed from simultaneously sampled time waveforms using the system ADC. Twelve-channel sine control requires a multiplexer and includes peak select in notching capability.

Figures 14, 15 and 16 show post-test plots of random, sine and transient tests run on a structure having the acceleration-to-amplifier voltage, transfer function shown in Figure 17.

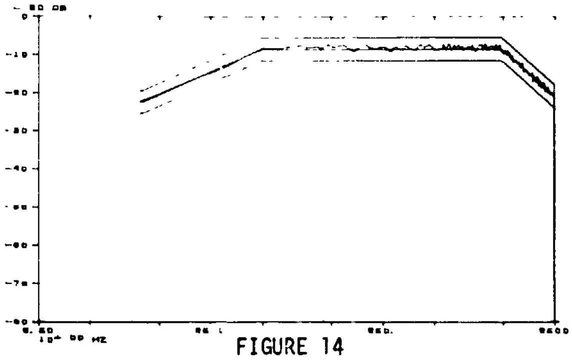


FIGURE 14

Random Control Spectrum with
Test Limits Superimposed

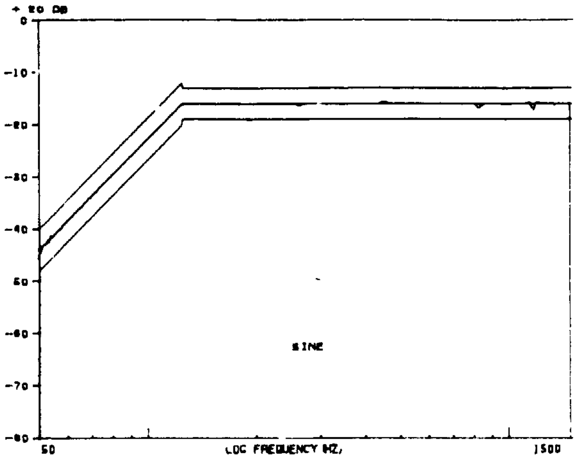


FIGURE 15

Sine Control Envelope with
Test Limits Superimposed

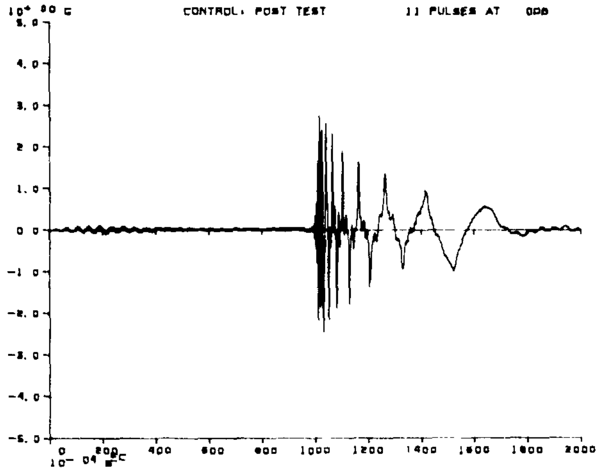


FIGURE 16

Transient Waveform

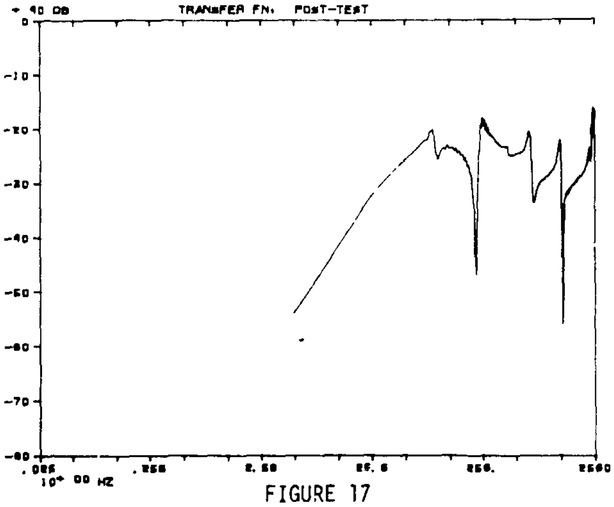


FIGURE 17

Transfer Function of Device Under Test

Perhaps the most remarkable contribution of digital vibration control is the time saved conducting tests. All of the "knobs" are set by operator responses to questions, and these test parameters can be recalled and set up in a matter of seconds. The high accuracy and repeatability of digital vibration control greatly simplifies test operation by, in most cases, eliminating "equalization runs" on bare fixtures.

Some of the other advantages of digital vibration control over conventional analog techniques are increased resolution and dynamic range in random tests; extremely flexible test strategies including multiple-channel notch control in sinusoidal testing; and shock spectrum analysis and synthesis in transient waveform control.

Some Management Considerations

A modern digital instrument/system is a truly powerful tool in the laboratory, but if it is to be used to its full capabilities it must be managed by people who understand the capabilities and limitations and who have the skills to implement the solutions to their unique measurement problems. This represents a cost after initial purchase which is sometimes overlooked and may result in underutilization of a valuable resource. The written documentation and training provided by the manufacturer are an important part of what is purchased. Therefore, the maintenance of the software and manuals in an organized and systematic manner is necessary to protect the investment.

Since most of the measurements listed in Table I can be performed using the keyboard programming language, there is often little need for the user to write special software. However, the capability is there. If it is used, careful documentation is required to make the results useful to others.

CONCLUSION

The broad range of measurements that can be made, coupled witn the vibration control capabilities, make a digital Fourier transformer-based instrument/system a necessity in a modern vibration laboratory. When properly managed and utilized, production can increase many times on routine work. Valuable measurements that used to be impractical, and were overlooked, can now be made.

REFERENCES

(1) Snyder, David; "Computer Performance Improvement by Measurement and Microprogramming," Hewlett-Packard Journal, February , 1975, Vol. 26, No. 6

(2) Hewlett-Packard, "Dynamic Testing of Mechanical Systems Using Impulse Testing Techniques," Hewlett-Packard Application Note 140-3, September 15, 1972

(3) Richardson, Mark, "Modal Analysis Using Digital Test Systems," Seminar Proceedings, "Understanding Digital Control and Analysis in Vibration Test Systems," Goddard Space Flight Center, Greenbelt, MD, June 17-18, 1975

(4) Richardson, Mark; Potter, Ron, "Identification of the Modal Properties of an Elastic Structure from Measured Transfer Function Data," 20th International Instrumenta-

tion Symposium, May 21-23, 1974, Albuquerque, New Mexico, Instrument Society of America #74250 (239-246)

(5) Potter, Ron; Richardson, Mark, "Mass, Stiffness and Damping Matrices from Measured Modal Parameters," International Instrumentation-Automation Conference and Exhibit, New York City, October 28-31, 1974, Instrument Society of America Paper #74-630

(8) Tebbs, J. Daniel; Hunter, Norman F. Jr., "Digital Control of Random Vibration Tests Using a Sigma V Computer," Institute of Environmental Sciences, 1974 Proceedings

(7) Witte, A.F.; Smallwood, D.O., "A Comparison of Shock Spectra and the Least Favorable Response Techniques in a Transient Vibration Test Program," Institute of Environmental Sciences, 1974 Proceedings

(6) McKinney, H. Webber, "Band Selectable Fourier Analysis," Hewlett-Packard Journal, April, 1975, Vol. 26, No. 8